A MODELING PROCESS TO UNDERSTAND COMPLEX SYSTEM ARCHITECTURES

A Thesis
Presented to
The Academic Faculty

by

Santiago Balestrini Robinson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2009

A MODELING PROCESS TO UNDERSTAND COMPLEX SYSTEM ARCHITECTURES

Approved by:

Professor Dimitri N. Mavris,
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor Brian J. German
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Janel Nixon
President and Owner
Integrative Engineering, Inc.

Professor Carlee A. Bishop
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 6 July 2009

For my wife, Rebecca.

ACKNOWLEDGEMENTS

I consider myself extremely blessed in that throughout my life I've always enjoyed the company and wisdom of extremely kind, smart, and strong people. I see this dissertation to be as much a result of their actions as I see it being a product of my labors. If I were to acknowledge all by name and explain what I've learned or received from each one, this section would easily surpass in length the remainder of this document. Therefore, let me make it clear that this is a synopsis.

My mother always said that an education is the most valuable thing a person can have. She not only said this repeatedly, she went to great lengths to make sure that my siblings and I got the best education possible. It is thanks to her that I find myself here today writing this acknowledgement section. It is thanks to her that I could pursue my dream and come to the United States to study that which I had been saying I wanted to study since I was 5 years old. For that I will be forever thankful, and forever in her debt. Thank you for letting me pursue my dream.

My father was *the* engineer in the family. When I first began looking at universities to study aircraft design, I was looking at simply that, design. There are very few universities that offer degrees solely on design. Most universities offered full engineering degrees, and design was considered a graduate specialization, even then, design was heavily reliant on the *hard sciences*. I had not thought of engineering seriously at first–being drawn more strongly by the art of design than the science–but now that I think about it, I am surprised I did not. My father was an engineer by trade, but I'm convinced that he would have been happier as an architect, or an artist for that matter. I'll always remember the multitude of maquettes and croquis, each one an almost perfect balance between aesthetics and function. I learned from him that good design must have an inherent beauty, and that the engineer has much to learn from art. And even though we may be lost in mathematical functions, numbers, books and scientific theories, there is still a place for art in engineering. I would

now say that this is particularly true of complex systems engineering. My father's lessons may not have been straightforward, but their value was immense, for all of them, I thank him.

I have one more "parent" to thank in this section. Bambina has taught me plenty, certainly more than I can write, but the most important thing I learned from her is that one must not falter when faced with adversity. Bambina's life has been harder than anyone else I know, yet she has a spirit that can only be described as inspirational, I cannot express in words how much I love and respect this second mother of mine. Grazie Bambina.

In all my travels I've always been fortunate to meet great people. My friends at Georgia Tech are no exception. I have to make a special mention to Hernando "Nando" Jimenez, my first and best friend here at Georgia Tech. Not only my best man, and companion in good times, Hernando's academic "carrier" and mine have been almost identical twins. Without his insight and companionship I would most likely not find myself where I am today. Other great individuals helped me with the ideas contained in this dissertation, they are, in no particular order, Ian Stults, Kelly Griendling, Daniel Cooksey, William Engler, Bjorn Cole, and all of you who attended my presentations and offered valuable insights and concerns.

I have to extend a very special thanks to my advisor, Professor Dimitri "Doc" Mavris. He was not only my advisor and boss, but a philosopher guide that by carefully phrasing questions, taught me more about science, engineering, and what entails to be a *doctor of philosphy* than what I could have learned by myself in a lifetime of academic wandering. His lessons spanned well beyond the academic, his insights into the business of research have been invaluable, yet I know there is much to learn still

The rest of my committee members went to great lengths to guide this dissertation from murky waters into what I hope are clearer ones. They have put invaluable time into my work, and for that I'll be forever thankful. Professor Schrage's great experience has been a great asset to have in this dissertation process, his willingness to take time from his busy schedule to help this graduate student has never ceased to impress me. Professor Bishop has been a great supporter and at the same time a great critic. Her commentaries have been infinitely appreciated and had made this work a much better one. Professor German may

have been the last committee member, but I can say that during his short time, he has put more time and effort than anyone could. The incredibly insightful comments, questions and recommendations have oftentimes left me at a loss for words, I cannot thank Brian enough. My last committee member was for all practical purposes my first. Dr. Nixon may have been a cubicle neighbor during my early years at ASDL, but she has demonstrated a deep understanding of the Ph.D. process and has provided me with innumerable ideas. Thank you all for taking the time and putting in the effort.

Last, but by no means last, I have to thank my wife. She has made as many (if not more sacrifices) for me to get my Ph.D. as I have. She has been both a rock of support and a whip of incentive, the perfect combination for a tenured student. I cannot put into words how fortunate I am to have this incredible woman by my side everyday. Thank you for it all darling!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

1GW First Generation Warfare

2GW Second Generation Warfare

3GW Third Generation Warfare

4GW Fourth Generation Warfare

AAM Air-to-Air Missile

ABM Agent-based Modeling

ABS Agent-based Simulation

ABM&S Agent-based Modeling & Simulation

ACS Autocatalytic Set

AF Air Force

AI Artificial Intelligence

AIC Algorithmic Information Complexity

AIMS APL Integrated Multi-warfare Simulation

AMN Assortative Mixing Network

ANOVA Analysis of Variance

APL Applied Physics Lab

APOD Air-Port of Destination

ASM Air-to-Surface Missile

ATCAL Attrition Calibration

BA Barabási-Albert

BFEM Battle Force Engagement Model

BSE Battle Space Entity

C2 Command and Control

C3 Command, Control, and Communications

C4ISR Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance

CA Cellular Automata

CASABA Critical-Aggregate Sensitivity Analysis-Based Algorithm

CCD Central Composite Design

CFM Capability-Function Matrix

CIC Combat Information Centers

CLM Causal Loop Models

CM Cognitive Maps

CNE Coefficient of Networked Effects

CNO Chief of Naval Operations

COA Course of Action

COFM Correlation of Forces and Means

CRN Common Random Numbers

CPN Colored Petri Nets

CS4 Combat Service Support Supply System

DAC Digraph Algebraic Connectivity

DAL Distributed Adaptive Logistics

DeMO Discrete-event Modeling Ontology

DoD Department of Defense

DoDAF Department of Defense Architecture Framework

DE Differential Equations

DES Discrete Event Simulation

DEVS Discrete Event System Specification

DIADS Digital Integrated Air Defense System

DIKW Data, Information, Knowledge, and Wisdom

DiMA Digraph Modeling for Architectures

DMAS Distributed Multi-Agent Systems

DO Distributed Operations

DOE Design of Experiments

DNO Distributed Networked Operations

DNOsim Distributed Networked Operations Simulator

DNW Dynamic Network

DTRA Defense Threat Reduction Agency

EAMA Executable Architecture Methodology for Analysis

EADSIM Extended Air Defense Simulation

EBM Equation-based Modeling

EBO Effects Based Operations

eDNOsim Enhanced Distributed Networked Operations Simulator

EGM Engagement Generation Matrix

EINSTein Enhanced Isaac Neural Simulator Tool

EM Engagement Matrix

F2T2EA Find, Fix, Track, Target, Engage, Assess

FCS Future Combat System

FIFO First-In First-Out

FIPS Federal Information Processing Standard

FIRE FLAMES Interactive Runtime Executable

FFT Fast Fourier Transform

FLAMES FLexible Analysis, Modeling, and Exercise System

FLARE FLAMES Analysis and Reduction Environment

FLASH FLAMES Scenario Highlighter

FNR False Negative Rate

FORGE FLAMES Operational Requirements Graphical Editor

FPR False Positive Rate

FSM Force Structure Matrix

FV Fiedler Vector

GCAM General Campaign Analysis Model

gDNOsim Generalized Distributed Networked Operations Simulator

GDP Gross Domestic Product

GIGO Garbage-In Garbage-Out

GPS Global Positioning System

GST General Systems Theory

GUI Graphical User Interface

GWOT Global War on Terror

HLA High-Level Architecture

HMMWV High Mobility Multipurpose Wheeled Vehicle

HMR Holistic Modeling Regret

ICM In-degree Constraint Matrix

IDEF0 Integrated Definition 0

IDS Integrated Deepwater System

IEEE Institute of Electrical and Electronics Engineers

IEH Individual Event History

IM Influence Models

INCOSE International Council on Systems Engineering

IOL Inter-Operability Level

ISAAC Irreducible Semi-Autonomous Adaptive Combat

ISR Intelligence, Surveillance, and Reconnaissance

IT Information Technology

ITEM Integrated Theater Engagement Model

JCIDS Joint Capabilities Integration Development System

JICM Joint Integrated Contingency Model

JIMM Joint Interaction Mission Model

J-UCAS Joint Unmanned Combat Air System

JWARS Joint Warfare System

LIFO Last-In First-Out

M&S Modeling and Simulation

MAD Mutual Assured Destruction

MCS Monte Carlo Simulation

MDST Missile Defense Space Tool

MOE Measure of Effectiveness

MOFE Measure of Force Effectiveness

MONE Measure of Network Effectiveness

MOP Measure of Performance

MRM Multi-Resolution Modeling

MSG Models, Simulations, and Games

MTO Major Theater of Operations

MTW Major Theater War

NATO North Atlantic Treaty Organization

NCO Network Centric Operations

NCW Network Centric Warfare

ND Non-Dimensional

NGATS Next Generation Air Transportation System

NM Network Model

NN Neural Network

NPS Naval Postgraduate School

NSF National Science Foundation

NSS Naval Simulation System

OCM Out-degree Constraint Matrix

OdC Off-diagonal Complexity

OEC Overall Evaluation Criterion

OFT Office of Force Transformation

OIF Operation Iraqi Freedom

ONR Office of Naval Research

OODA Observe, Orient, Decide and Act

OOTW Operations Other Than War

OPM Object-Process Methodology

OR Operations Research

OSD Office of the Secretary of Defense

PFE Perron-Frobenius Eigenvector

PPM Pathway Participation Metric

QJMA Quantified Judgment Method of Analysis

RBN Random Boolean Network

RCS Radar Cross Section

R&D Research and Development

RMR Reductionist Modeling Regret

RMT Random Matrix Theory

SAD Simulation Activity Diagrams

SAM Surface-to-Air Missile

SAMS Surface AAW Multi-ship Simulation

SAS Special Air Service

SBE Simulation-based Engineering

SBS Special Boat Service

SC Stochastic Compartmental

SD System Dynamics

SEAS System Effectiveness Analysis Simulation

SFM Stock-and-Flow Model

SFN Scale-Free Network

SFR Straight Flush Radar

SIR Susceptible-Infected-Recovered

SOC Self-Organized Criticality

SOF Special Operation Forces

SoS System-of-Systems

SPOD Sea-Port of Destination

SPN Stochastic Petri Nets

SS Schutzstaffel

SSC Smaller-Scale Contingency

STORM Synthetic Theater Operations Research Model

SysML Systems Modeling Language

TACWAR Tactical Warfare

TNDM Tactical Numerical Deterministic Model

TOGAF The Open Group Architecture Framework

TSM Time State Matrix

UAV Unmanned Aerial Vehicle

UCAV Unmanned Combat Air Vehicle

UCS UAV Control Station

UML Universal Modeling Language

USA United States of America

USAF United States Air Force

USG United States Government

USSR Union of Soviet Socialist Republics

V&V Verification and Validation

WMA Warfare Mission Area

WWI World War I

WWII World War II

SUMMARY

Analysis–the basis for rational decision making–relies on models. Models are, in the most generic sense, abstractions that are useful representations of reality. Engineers tend to rely on mathematical models–generally in the form of formulas that relate some property of interest to some property that can be manipulated–but these are just one of the many types of models that exist. The equation-based models have served science and engineering well. The ability to study a problem, identify the characteristics that are critical and drive its behavior, and relate those to the behaviors of interest has been the focus of much of science. Problems of simplicity, those characterized by a few entities interacting in a complicated manner, and disorganized complexity, those characterized by a large number of entities interacting in a simple manner, as described by Warren Weaver [338] have become well understood by science. The new realm of interest is in between these two areas, the problems of organized complexity, what has now become known as *complexity science*, the science of emergence, that multidisciplinary amalgamation of concepts which attempts to understand the intrinsic characteristics of *the complex*.

Engineers and designers are pursuing applications that fall within the realms of *complex systems*, or are designing systems to operate within the scope of larger aggregations of systems for which they have no direct control. The goal is then to shape the behavior of the larger systems by introducing changes in their interactions and the systems that compose them. Examples abound, but this is particularly true of the U.S. military, with its requirement to move away from threat-based analysis to force-level capability-based assessments, pursuing not the design of independent systems, but of Systems-of-Systems, and Families-of-Systems, envisioning Network Centric Operations where the force is a network of distributed functionality, with higher levels of Jointness and distributed Command and Control. These aggregations of systems, referred to as large-scale system architectures,

tend to display characteristics of complex systems. They are complex in their composition (they are composed of a large number of systems which interact nonlinearly) and in their behavior (they display emergent behaviors). At the same time, the change in focus (e.g., network-centricity, higher levels of force jointness, distributed command and control, etc.) has moved design of the military systems away from the physics and into newer realms, e.g., human-machine interface design, design for emergence, etc. The methods employed for evaluating candidate systems, doctrines, and technologies (e.g., capability-based acquisition, a focus on architectures rather than systems, etc.) is shifting the design process in itself. Why test the ability of a system to perform in a certain way when the goal is to test if a new system is indeed needed? When the goals are not clearly specified in engineering terms, but stipulated as abstracted objectives, the work of the analysts increases to the point of being unmanageable. In recent decades, several tools have been developed by the armed forces, and their contractors, to test the capability of a force. These campaign level analysis tools, often times characterized as *constructive simulations* are generally expensive to create and execute, and at best they are extremely difficult to verify and validate.

This central observation, that the analysts are relying more and more on constructive simulations to predict the performance of future networks of systems, leads to the two central objectives of this thesis: (1) to enable the quantitative comparison of architectures in terms of their ability to satisfy a capability without resorting to constructive simulations, and (2) when constructive simulations must be created, to quantitatively determine how to spend the modeling effort amongst the different system classes. An extensive literature search of modeling and simulation, led to the narrowing of four candidate techniques, network modeling, discrete event simulation, system dynamics, and agent-based modeling. The level of fidelity increased from the former to the latter, but the ease of creation and execution decreased as well.

The first objective led to Hypothesis A, the first main hypotheses, which states that by studying the relationships between the entities that compose an architecture, one can infer how well it will perform a given capability. The method used to test the hypothesis is based on two assumptions: (1) the capability can be defined as a cycle of functions, and

that it (2) must be possible to estimate the probability that a function-based relationship occurs between any two types of entities. If these two requirements are met, then by creating random functional networks, different architectures can be compared in terms of their ability to satisfy a capability. In order to test this hypothesis, a novel process for creating representative functional networks of large-scale system architectures was developed. The process, named the Digraph Modeling for Architectures (DiMA), was tested by comparing its results to those of complex constructive simulations. Results indicate that if the inputs assigned to DiMA are correct (in the tests they were based on time-averaged data obtained from the ABM), DiMA is able to identify which of any two architectures is better more than 98% of the time. DiMA was developed with the current DoD guidance in mind, and for that reason, it was intended to leverage as many of the existing DoDAF products as it would be necessary in order to ease the creation of the input data sets and reduce the amount of rework required from the analyst.

The second objective led to Hypothesis B, the second of the main hypotheses. This hypothesis stated that by studying the functional relations, the most *critical* entities composing the architecture could be identified. The *critical* entities are those that when their behavior varies slightly, the behavior of the overall architecture varies greatly. These are the entities that must be modeled more carefully and where modeling effort should be expended. This hypothesis was tested by simplifying agent-based models to the non-trivial minimum, and executing a large number of different simulations in order to obtain statistically significant results. One ranking was proposed as being the best at prioritizing modeling effort, and compared to 13 other ones, including a uniform ranking to test the sub-hypothesis that ranking is beneficial, and a random ranking to test the sub-hypothesis that *intelligent* ranking is beneficial. These rankings were not based on the dynamic behavior of the model, their input was solely the functional structure of the architecture. For the testing, modeling effort was assumed to be inversely proportional to modeling error, due to the fact that there is a higher probability of capturing an erroneous behavior in a system if little effort is expended in modeling it, and viceversa. The tests were conducted by evolving the complex model without any error induced, and then evolving the model once again for each ranking

and assigning error to any of the nodes with a probability inversely proportional to the ranking. The results from this hypothesis test indicate that depending on the structural characteristics of the functional relations, it is useful to use one of two of the intelligent rankings tested, or it is best to expend effort equally amongst all the entities. Random ranking always performed worse than uniform ranking, indicating that if modeling effort is to be prioritized amongst the entities composing the large-scale system architecture, it should be prioritized intelligently. The benefit threshold between intelligent prioritization and no prioritization lays on the large-scale system's chaotic boundary. If the large-scale system behaves chaotically, small variations in any of the entities tends to have a great impact on the behavior of the entire system. Therefore, even low ranking entities can still affect the behavior of the model greatly, and error should not be concentrated in any one entity. It was discovered that the threshold can be identified from studying the structure of the networks, in particular the cyclicity, the Off-diagonal Complexity, and the Digraph Algebraic Connectivity.

CHAPTER I

INTRODUCTION

> "You have a choice: you can either create your own future, or you can become the victim of a future that someone else creates for you. By seizing the transformation opportunities, you are seizing the opportunity to create your own future."
>
> - VADM Arthur K. Cebrowski

If one were to dissect the title of this thesis, several terms would need to be described in more detail and explained in the proper context to elucidate the purpose of this body of work. For this reason, we will begin by describing what is meant by "A Modeling Process to Understand Complex System Architectures." The word *modeling* simply refers to the creation of models, models are ubiquitous to every activity we perform, and can be meant to signify computer models, mathematical models or cognitive models, that is, models that reside in the mind. Models are necessary to help us understand the relationship between the causes and effects of the reality that surrounds us and to attempt to predict the behavior of the systems we observe and interact with. The reason why it is a modeling *process* and not simply a model is that no one model will properly represent every possible application. The expertise of the disciplinarians is still required to develop the models and tailor them to the appropriate application. This thesis is meant to provide a proven guideline, a *process*, to achieve that goal. The word *understand* signifies that the purpose for which these models are to be constructed is not necessarily to operate on them (e.g., optimize), but to help elucidate the cause and effect relations and answer questions of the *why* form. The definition of understanding and how it relates to concepts like data, information and knowledge will be discussed in more detail in the background section. Throughout the last few decades, the term *complex system* has been extensively used, and often misused, for different purposes. A more thorough definition of complexity will be provided further on,

but for the purposes of clarification the main characteristics of a complex system are that of a functionally related group of elements (a system) that exhibit nonlinear (disproportionate cause-to-effect), interdependent (the elements are related to each other in a variety of ways) and emergent behaviors (the macroscopic behavior of the system cannot be determined from the sole analysis of the individual, microscopic, behaviors of the elements). Finally the word *architecture* signifies that this thesis is meant to support the understanding of more than just the structural elements but also their interfaces, functional assignments and distributions. In other words, this thesis is meant to be a proven guideline (process) for creating tools (models) that help us elucidate the cause-and-effect relations and behaviors (understand) of arrangements of elements, along with their interfaces, assignments and distributions (system architectures) for which we are not very adept at (complex).

Although the motivation for this thesis is ignited by the transformation in the conduction of warfare, the methods that will be developed are not exclusively applicable to complex military architectures but rather to complex architectures in general, e.g., the Next Generation Air Transportation System (NGATS) [277]. Problems outside of engineering -e.g., epidemiology [200], economics [44], sociology [253, 44], electric power distribution [27, 113], national security [19], combat analysis [173], etc.- could also benefit from these approaches and techniques.

1.1 The Evolution of Warfare

> "At an earlier time, a commander could be certain that a future war would resemble past and present ones. This enabled him to analyze appropriate tactics from past and present. The troop commander of today no longer has this possibility. He knows only that whoever fails to adapt the experiences of the last war will surely lose the next one."
>
> - Gen Franz Uhle-Wettler
>
> German Army [216]

If the history of warfare since World War II (WWII) is revisited we can see that we

are undergoing a dramatic change in the way we conduct warfare. The advent of large-scale coordinated mechanized maneuver warfare observed during WWII, where the offensive tactics were to penetrate the enemy lines and jeopardize their logistical supplies while the defensive tactics were to defend "in depth" to avoid this, revolutionized the manner in which warfare was conducted. Not only the tactics developed for previous conflicts became obsolete, but the models used to understand conflict became obsolete as well. No longer did the attrition tactics nor models satisfy this new paradigm. The Germans and the Soviets were the first to realize this by utilizing coordinated air and fast moving mechanized units to penetrate enemy lines and strike deep into enemy territory. These tactics allowed Nazi Germany to make unprecedented advances in its attack on their Eastern and Western fronts. World War II was a horrific conflict to a scale never seen before nor since that changed the world forever and revolutionized the conduction of warfare. With the end of WWII in 1945, a new balance of global power emerged which gave birth to the Cold War and saw the blooming of the Nuclear Age.

Two superpowers with conflicting ideologies emerged from the ashes of World War II. On one side stood the technologically adept and economically healthy United States of America (USA) and on the other the Union of Soviet Socialist Republics (USSR), which adopted a quantity over quality approach to bridge the technological gap with the USA. These conflicting sets of strategies were equilibrated by the advent of nuclear weapons and the massive development of these along with revolutionary delivery methods to equilibrate gaps in technology, numbers, strategies and tactics. This led to a new symmetrized stable paradigm of warfare characterized by the Mutual Assured Destruction (MAD) doctrine. Nevertheless, during this period, each of these superpowers suffered a defeat to the hands of what can be considered to be *lesser* enemies. For the US was the North Vietnamese and the Vietcong guerrillas between the years of 1959 and 1975. For the Soviets it was the Islamic and Afghan guerrillas between 1979 and 1988. These two "David versus Goliath" conflicts were characterized by superpowers attempting to use large-scale conventional forces against a less technologically capable, yet more maneuverable and adaptable enemy. Conventional in this thesis will only refer to forces not employing radiological or nuclear weapons, and

by no means should be associated to the use of guerrilla or irregular tactics.

The terrible loss suffered by the USSR in Afghanistan, along with other reasons like the collapse of its economy, propelled its dissolution in 1991 and initialized what can be characterized as the period of "American Hegemony." At that point in time no nation on Earth could challenge the United States neither militarily or economically. During this period the USA was involved in a series of conflicts and Operations Other Than War (OOTW), the first of which was Operation Desert Storm. Most of the military campaigns during this period, and in particular Operation Desert Storm, proved to be a rotund success, reverting many of the impressions and fears from Vietnam and provided great confidence in the technological superiority of the United States. All this would be challenged by the sudden terrorist attacks of September 11, 2001, which signified the transition point to a new era of global conflict.

The conflicts during this period have come to approximate the conventional conflicts of the Cold War, e.g., Vietnam, Afghanistan, etc., where large conventional forces occupy an adversary's territory for extended periods of time in an attempt to bring stability to a region. During this time, the tactics of the adversary adapt to the occupying force and exploit its larger inertia to equilibrate the disparity in power, technology and numbers.

This brief recounting of the history of the last 60 years of warfare provides a glimpse at how warfare is evolving, more importantly, how the paradigm of the conduction of warfare is evolving. A paradigm shift, as initially described by Kuhn [202], is occurring because the basic assumptions on the ruling theories, in this case on warfare and how it should be conducted, are drastically changing.

1.2 Paradigms of Warfare

> "Throughout history, warfare has assumed the characteristics of its age and the technology of its age. Today we see this trend continuing as we move from the Industrial Age warfare with its emphasis on mass to Information Age warfare, which highlights the power of networked distributed forces and shared situational awareness."

- Paul Wolfowitz, July 2001

Former U.S. Deputy Secretary of Defense [79]

This quote by former Deputy Secretary Wolfowitz illustrates very clearly the driving force behind the current evolution in the paradigm for the conduction of warfare. In this section the two paradigms will be characterized and contrasted.

1.2.1 Industrial Age Warfare

The main characteristic of Industrial Age Warfare is that it used first mass production and then higher mobility, or maneuver, to produce more agile forces that demand defense "in depth" because the offense attempts to cut through the enemy's lines and strike deep into enemy territory to disrupt its supply lines.

With the advent of the machine gun prior to World War I (WWI), the balance of warfare was tilted towards defense, with many military theorists arguing that the stank battles of WWI would become the norm in war [321]. It was not until the advent of the tank, the troop carrier, the attack plane, and more importantly, the portable radio, that the higher mobility engagements characteristic of WWII could be conducted. The portable radio was key in enabling the sharing of information (both in terms of orders and awareness) which was the backbone of strategies like the German *Blitzkrieg* and the Soviet *Deep Operations* [298, 159].

Nevertheless, Industrial Age Warfare retained some of the ideologies of trench warfare in that it remained linear in form, in other words, with the exception of a relatively small number of operations the networking of forces was considered to be secondary to numbers (mass), the whole was still considered to be approximately the sum of its parts. On the other hand, great strives were undertaken by small groups as for example the British Special Air Service (SAS) and Special Boat Service (SBS) and Otto Skorzeny of the Waffen Schutzstaffel (SS). The SAS was born in 1941 in the deserts of North Africa from ideas laid out by CAPT David Stirling. His main hypothesis was that many small fast-moving forces could jeopardize the enemy's situation more than large slow moving ones, this was demonstrated in the engagements of North Africa between 1941 and 1942 [245, 125]. Paralleled to the

SAS, the SBS, evolved from lessons learned from the Italian commando frogmen captured between 1941 and 1942. The SBS saw combat in a wide range of theaters, e.g., Northwestern Europe, Norway, the Middle East, France, Italy and Burma [219]. The British were not the only to recognize the advantages of employing distributed, highly specialized forces. The Nazis under the guidance of Otto Skorzeny led the operation to liberate Benito Mussolini against impossible odds from a British prison in Italy [174], trained the "American Brigades" who infiltrated allied lines after the invasion of Normandy during the Battle of the Bulge, and helped train the Werewolves, special guerrilla groups ordered to fight the allies after the fall of Nazi Germany [299]. Regardless of the effectiveness of these special groups and the effect they had on the war, they conducted a very small percentage of the total number of operations and did not radically change the tactics of the regular forces.

> "Defense planning during the Cold War was dominated by the threat from the Soviet Union. It was, in that sense, threat based. It also was, to a great extent, symmetrical, based on force-on-force calculations for U.S. and Soviet armored forces, fighter jets, and the like. In these circumstances, the U.S. planning structure within the Pentagon became increasingly centralized, seeking to maximize the benefits from various investments in ways to better cope with the Soviet threat.
>
> All the practices that made considerable sense during the Cold War badly need to be rethought now. Soviet strategy may have been more creative than it was usually given credit for, but it was relatively slow moving. By contrast, todays threatsand still more tomorrowsare many and very uncertain. While none may be in a class with the Soviet threat, the attacks of September 11, 2001, drove home how lethal even lesser threats can be. Moreover, U.S. military power has given rise to a paradox: the United States is so dominant in its ability to fight a conventional armored war that it is not likely to have to fight such a war. Realizing the futility of a conventional face-off with the United States, would-be adversaries will instead aim to confront the United States where it is weak or can be surprisedposing what are called asymmetric threats. Terrorism,

the strategy of the weak against the strong, is quintessentially an asymmetric strategy."

Johnson et. al 2003 ([183], pp. 10)

Despite the fact that these operations constituted a small minority of the operations of the war, the lumping of WWI and WWII under the category of Industrial Age Warfare may be too broad for the purpose of understanding the evolving characteristics of warfare. In their seminal 1989 paper "The Changing Face of War: Into the Fourth Generation," [216] Lind et al. characterize Industrial Age Warfare as a blend of what they define to be second and third generation warfare. These are paradigms that rely on mass and maneuver respectively, WWI being described as Second Generation Warfare (2GW) and WWII as Third Generation Warfare (3GW). The fast coordinated mechanized-air maneuver used by the Germany in WWII, as mentioned before, was enabled by the portable radio, that allowed for the use of shared information and faster turn-around times (feedback). Lind et al. used the lessons learnt from their characterization of warfare to predict what they define to be Fourth Generation Warfare (4GW). Nevertheless, they recognize that maneuver will be a critical component of fourth generation warfare, a concept that parallels and can be encompassed within information age warfare. Despite these similarities between Information Age Warfare and Industrial Age Warfare, radical differences remain. Regardless of the use of maneuver by both, maneuver takes two completely different forms and scale. Despite the fact that 3GW applies some of the concepts of Information Age Warfare, (e.g., Blitzkrieg) Information Age Warfare will demand coordination and distribution of forces to a scale not previously seen in the history of conflict. Furthermore, these engagements are still characterized by symmetric, force-on-force, large-scale battles using tactics of infiltration to bypass and collapse the enemys combat forces, and the defensive technique still relies on defense in depth 3GW. Command is highly hierarchical, with a reliance in centrally coordinated control, leading to feedback that is in the order of days and rigid protocols that do not allow it to adapt as fluidly as what is envisioned for Information Age Warfare.

1.2.2 Information Age Warfare

Information Age Warfare is characterized by asymmetric, distributed small-scale engagements. It is distributed in nature, meaning that the units are dispersed and have a higher degree of control. This enables the forces to be adaptive but it requires them to rely on self-synchronized control. Feedback is expected to be in the order of minutes, an order of magnitude faster than Industrial Age Warfare. In some circles, it is often related to 4GW, even though it is not necessarily a 1-to-1 relation since 4GW is meant to signify a blurring of warfare and politics, while Information Age Warfare is meant to signify a more integrated, distributed, and emergent form of warfare. The enabling concepts for conducting warfare have been the U.S. Navy's Network Centric Warfare (NCW), or Network Centric Operations (NCO) [79], the U.S. Marine Corps' Distributed Operations (DO) [321], the U.S. Army's Future Combat System (FCS) [230], the U.S. Air Force's Transformation Flight Plan 2004 [280], and to some degree, the U.S. Coast Guard's Integrated Deepwater System (IDS) [9].

As mentioned previously, the motivation for these enabling concepts has been threefold. On the one hand there has been an evolution of the threat, i.e., Islamic Fundamentalists replaced the Warsaw Pact after the dissolution of the USSR [18]. The environments in which the future soldiers of America will need to fight are more complex, e.g., an interest in conducting littoral operations over traditional blue water ones [43]. Globalization and the internationalization of operations forces tighter integration with disparate technologies, tactics and protocols, (e.g., global security efforts [25]. Finally, the advance of technology, effectively enables new concepts to emerge (e.g., demand logistics networks replacing traditional supply chains [45, 73]) and are forcing the Department of Defense (DoD) and its branches to reconsider what is possible or best for the future warfighter.

The term "information age warfare" and its synonyms have been used extensively by a number of authors who attempted to characterize it, analyze it, and understand it. The Office of Force Transformation (OFT) guided by VADM Cebrowski formulated a strategic guide to Information Age Warfare [78] in which they describe it as part of containing three domains: the cognitive (the mind), the information, and the physical as depicted in Figure

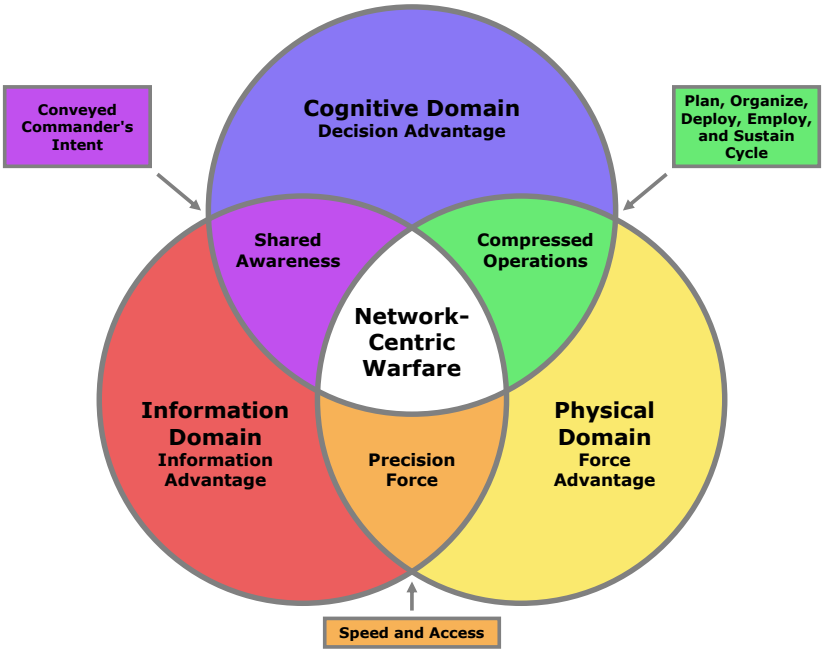

Figure 1: Domains of Conflict of Information Age Warfare. ([78], p. 33)

1. They describe these domains as the following.

Cognitive Domain This domain is in the mind of the warfighter. It is associated more closely with Effects Based Operations (EBO) which will be described in more detail below. It is characterized by intangibles such as leadership, morale, unit cohesion, situational awareness, and level of training and experience. It contains a commanders intent, doctrine, tactics, techniques, and procedures.

Information Domain Is where information is created, manipulated, and shared. It facilitates the communication of information among warfighters. It is where Command and Control (C2) is communicated and the commanders intent is conveyed. This is where many of the recent efforts concentrate because the reliance on information is increasing and gathering, protecting it, and communicating it are some of the greatest challenges.

Physical Domain This is the traditional domain of warfare, it is where forces traverse

> time and space. In this day and age it spans land, sea, air, and space. It is not only where military forces execute operations, but also where the physical platforms and communication networks that they employ reside. The tangible nature of this domain makes it the easiest to measure, for this reason, traditional measures of effectiveness and combat power have been measured in this domain.

Since the release of this strategic vision, changes have taken place and the concepts have evolved. More recent sources, for example, have focused on the adaptation of the command and control architectures of future military forces [35, 217, 274] because this area is where the most noticeable immediate changes will occur. But the truth is that Information Age Warfare is not simply an adaptation of command and control. Technologies and doctrine have to support this transition in order for it to become a reality. Figure 2 describes the necessary steps in enabling what has been defined as Information Age Warfare, where the two main characteristics are self-organized command and control to maximize adaptability, and shared awareness amongst the forces to enable them to exploit situational knowledge. This figure is based on an adaptation by ([217], p. 153) of Figure 76 in ([20], p. 241). The goal is to incrementally reach the goal by traversing the states in the four steps presented. Each transition demands either technology (in the case of going up) and new doctrine (in the case of going right). For example, when the F-14A was introduced with its AWG-9 radar, the protocols for command and control were not updated. This meant that the superior awareness provided by the new technology was not exploited, as it was demonstrated in exercise *Beacon South*. During that exercise, F-14s were tasked with protecting the battle group's air defense from F-111s piloted by the Australian Air Force. The protocol dictated that the F-14s should follow commands and vectors from E-2s and ship Combat Information Centers (CIC) when available. During the beginning of the engagement, F-14s tracked the F-111s with their own AGW-9 radars, but were directed away to pursue a different set of nonexistent targets by a ship-based CIC controller. This experience was the basis for the introduction of "vector logic" and it influenced the development of the "outer battle," radically changing how air defense was to be conducted by battle groups ([21], p. 76). With the introduction of the technology, the awareness was possible, but it did not become

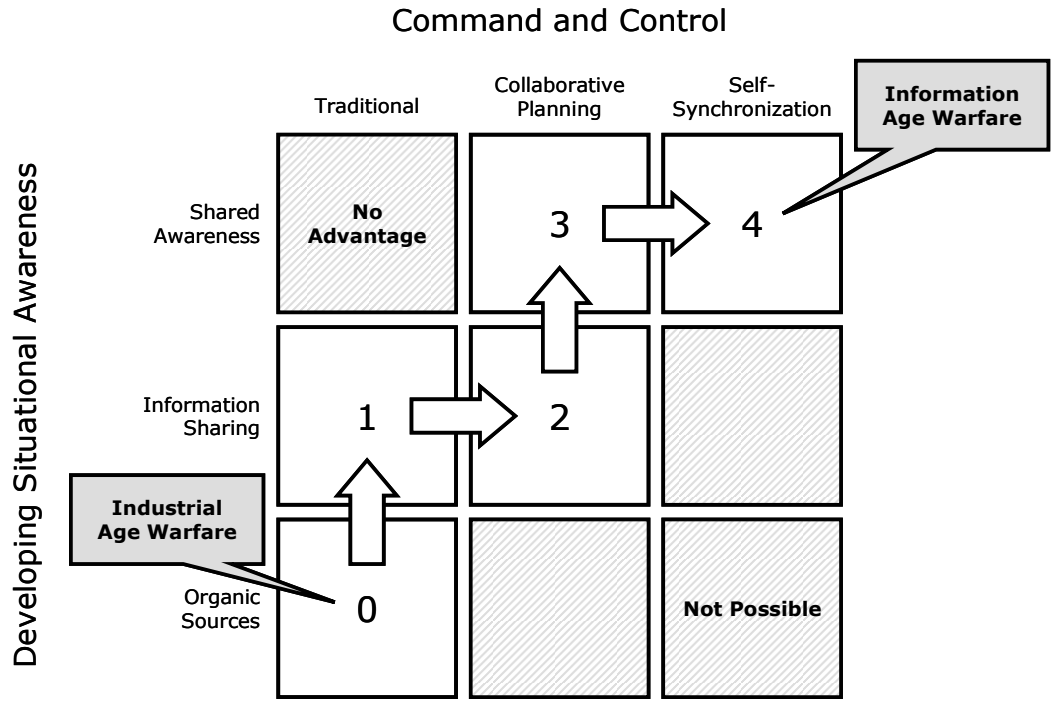

Figure 2: Network Centric Maturity Model. ([217], p. 153)

useful until the protocols for command were updated. Vector Logic formulated a method for creating shared awareness and allowing E-2 and F-14 platforms to share data to pin point the incoming bomber forces more accurately and reliably. To demonstrate how this example maps to Figure 2, the introduction of "vector logic" enabled the air defense system of the battle group to operate on the fringes (bottom-left corner if you will) of box 4. This is an encapsulated example, in reality, most of the military systems are at a level between 0 and 2. The shaded regions of the figure represent conditions that are either not favorable, as is the case of shared awareness with traditional top-down control, where it is not important to the warfighter to know what is the larger picture, because specific orders have been issued and the information critical for his actions was already available at the information sharing state. Or are not possible, as for example, collaborative planning cannot be done without information sharing, since in order to collaborate, warfighters must be able to communicate their conditions and intent.

Information Age Warfare requires adaptation of four dimensions: (1) the mission space, (2) environment, (3) organizational concept, and (4) the business concepts. Alberts ([300], p. ix) argues that EBO focuses on the first two, while NCW focuses on the last two. Mission space contains the set of tasks that the military is charged to do. This charter in essence, or at a high level of abstraction, has remained unchanged for centuries, i.e., defend the nation and its interests, but in the details is where the evolution has taken place. For example, the task of nation building in Afghanistan and Iraq have been questioned as true military tasks. At the same time, the environment, i.e., the constraints, conditions and values that affect the success of military operations, has been adapting. The legacy of Industrial Age Warfare, has been the linear, symmetric thinking that means equate to the effects. When confronting the USSR, North Atlantic Treaty Organization (NATO) could equate loss exchange ratios to determine how the war was being conducted. The same is not true of today's environment, where large casualties on the enemy's side do not imply that their capability to affect us adversely has been jeopardized. The advent of asymmetric engagements, where the "weaker" side mutates to level the playing field, demands flexibility to adapt. Militaries around the world are therefore transforming themselves to become more agile [35]. As the front lines become more blurry, and aggressors become non-state actors instead of nation states, there is a seemingly an increasing need to conduct operations that are neither exclusively civilian nor military. EBO attempts to address this complex need by studying what effects are desired in the ambiguous battlespace and how the success of said missions can be measured [300, 301]. The third and fourth dimension of the conduction of warfare, those addressed by NCW focus on the concepts that enable the branches of the DoD to achieve EBO and enable true Information Age Warfare.

While a radical change in capabilities and doctrine is required from the DoD, the fiscal support to enable this transition is diminishing. Figure 3 presents the military expenditures by the United States as percent of the Gross Domestic Product (GDP) over time. The critical feature of this chart is that the relative amount expended on the military is decreasing, and despite the fact that the GDP has increased over time sufficiently to increase absolute spending in defense, the fact that the portion expended on the military shrinks over time

is an indicator that it is harder to maintain an effective force. This is due to the fact that as a larger portion of the GDP is expended in other areas, the US Military faces increased competition for the same skilled labor force and resources. The data for the figure was obtained from information found in Table 3.1 [326] pp. 46-54.

1.2.3 Example: Military Logistics

A specific area of military operations where the concept of NCW is being readily applied is logistics. American military operations are often characterized as "overwhelming firepower supported by overwhelming logistics."[1] The importance of logistics is something that has been recognized by great military strategists going back to Napoleon, who is quoted to have said: "an army moves on its stomach," referring to the fact that armies need more than just ammunition to become an effective combat force. The importance of logistics cannot be overstated, yet today's logistics are performed in a very similar manner to the way they were performed 60 years ago. Even a campaign as Operation Iraqi Freedom (OIF), where unprecedented levels of information gathering, processing, and sharing, have enabled the USA to achieve spectacular results, the logistics that supported it are without a doubt the weak link. Examples abound, the inability to sustain large-scale operations without the use of a friendly Sea-Port of Destination (SPOD) or Air-Port of Destination (APOD) (e.g., the rerouting of the 4th Division), the vulnerable and tail-heavy in-theater supply chain (as demonstrated by the number of ambushed and kidnapped logistics convoys), the large amounts of supplies that must be prepositioned, and the delays in upgrading and maintaining the equipment (e.g., maintaining and up-armoring the HMMWVs in Iraq and Afghanistan) are all signs that the traditional supply chain is lagging behind the capabilities of the force it is supposed to supply and maintain. The following quote was made by a Lieutenant General in Afghanistan with regards to OIF.

> "Our logistics professionals achievements in OIF were especially spectacular in light of the fact that we supported a 21st century battlefield with a mid-20th century logistics structure."

[1] Unknown

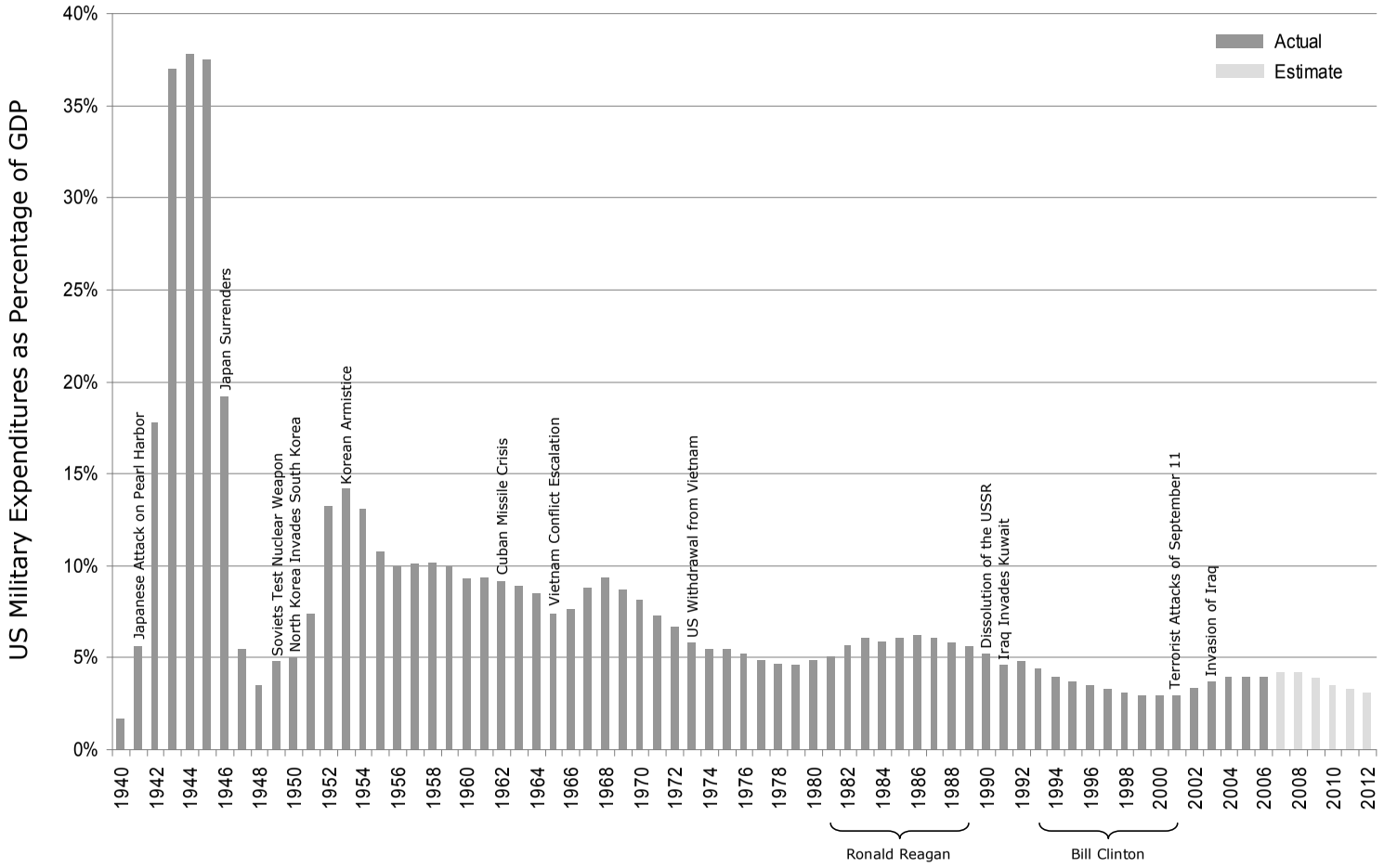

Figure 3: Military Expenditures as Percent of Gross Domestic Product. [326]

- Lt Gen C.V. Christianson, Baghram, February 2002 [244]

The application of information age logistics has taken many names during the recent years. One group has been centered around the argument for demand-pull versus supply-push and is commonly associated with the infusion of information technologies to the tracking and decision making processes involved in a logistical operation, e.g, Focused Logistics [45]. Others are facing the fact that these technologies can offer unprecedented capabilities if the doctrines and protocols are adapted as well [71, 254, 327]. The two groups have disparate basic goals, one is to make supply chains more efficient, the other looks into modifying the concept of logistics altogether. Of the latter, Jeff Cares [70, 71] is probably one of the main proponents of adapting the way people think of military logistics. He has evaluated the idea that self-organization can be used to make military logistics more adaptable and fluid and termed it Distributed Adaptive Logistics (DAL) [71]. The basic premise is that local commanders can learn local rules that will address global governing principles. This has profound implications to the way the military conducts warfare, it proposes a loser hierarchy that can adapt as the situation in the battle progresses.

1.3 The Impact on Design

The transition from Industrial Age Warfare to Information Age Warfare has been identified as a critical necessity for the United States. Two concepts, EBO and NCO, attempt to address the key areas of this transition. EBO are sets of actions directed at shaping the behavior of friends, neutrals and foes in times of war, crisis and peace.[91, 167] NCO are a set of guidelines by which capabilities are the product of interaction between systems and not just the systems themselves. These are concepts, but do not necessarily establish how the engineering must be performed.[79] The Manhattan project and the NASA Space Program are considered to be two of the most complicated large-scale engineering problems ever conducted. The main difference between these two pinnacles of engineering and the development of new complex large-scale architectures are the following ([42], p. iii):

1. Substantial technology will be needed to achieve the goals.

2. This technology is based on a clear understanding of the basic principles (e.g., $E = mc^2$ for the Manhattan project, and $F = \frac{-GMm}{r^2}$ for the Space program).

3. The goal and the objectives are clearly understood.

4. The new system will be developed from scratch.

These are assumptions that clearly do not apply to the engineering of information age military architectures. It is not necessarily a matter of developing advanced technology, but of changing doctrine and other intangibles, to effectively exploit technology appropriately. There is not a clear understanding of the principles that produce self-organized systems [182, 199]. The top level goals may be know, but not understood, the objectives are not agreed upon. Finally, the new system will not be developed from scratch, but evolved over time due to budgetary and training constraints. To add upon the divergence between these two ideal large-scale engineering projects, there is the fact that non-physical systems are more complex to be designed than physical ones ([133], p. 2). New tools are being developed, but there is still much to be done, in particular with understanding how the transformation must be conducted.

> "But really, this is precisely what transformation is all about. Here we are in the year 2002, fighting the first war of the 21st century, and the horse cavalry was back and being used, but being used in previously unimaginable ways. It showed that a revolution in military affairs is about more than building new high tech weapons, though that is certainly part of it. Its also about new ways of thinking, and new ways of fighting."
>
> - SECDEF Donald Rumsfeld, National Defense University
>
> January 31, 2002 ([78], p. 5)

This quote by the Secretary of Defense is in reference to the use of packing animals in Afghanistan to aid special forces. Currently, much of the innovation is taking place for the bottom up. The clear example are the special forces teams hunting Taliban groups in the mountains of Afghanistan. When the USA invaded Afghanistan in October 2001 it faced a

poorly equipped Taliban force that was no match to the United States Military in the open field. This forced the Taliban to take refuge in the highly inhospitable mountainous terrain in its border with Pakistan. Large scale forces were not suitable to fighting the Taliban since they lost the element of surprise. For this reason, the American armed forces relied heavily on Special Forces who could track and illuminate Taliban strongholds in the mountains, and have aircraft eliminate them with precision munitions. The difficult terrain made it difficult to supply these forces, especially since using helicopters would compromise their position during their long missions. Figure 4 depicts Special Operation Forces (SOF) riding with soldiers from the Northern Alliance in Afghanistan in late 2001. This sudden reliance on animals caught the armed forces by surprise, which had not used pack animals since the Korean War. A manual for operating with pack animals had to be compiled and released [99] to train forces, but it was not available until 2004. The forces made further use of ubiquitous technologies like off-the-shelf palm pilots to help them calculate coordinates provided by their laser designators and Global Positioning System (GPS). Satellite communications allowed them to call in B-1B Lancers who have the ability to loiter for long hours with large payloads of precision munitions. The implementation of these technologies, doctrinal adaptations to the kill chain, and tighter integration of assets have reduced the time to engage by an order of magnitude. But the approach so far has been *ad hoc* and leveraging whatever technologies are available at the time. Designers need to apply scientific methods to the development of these systems to reduce their cost, and improve their performance and reliability, effectively exploiting paths that may not be readily noticeable through the trial and error *ad hoc* approach currently employed.

> "The need for military transformation was clear before the conflict in Afghanistan, and before September the 11th. . . . What's different today is our sense of urgency the need to build this future force while fighting a present war. It's like overhauling an engine while you're going at 80 miles an hour. Yet we have no other choice."
>
> - President George W. Bush, The Citadel, Charleston, SC

Figure 4: Special Forces use of animals in Afghanistan

December 11, 2001 ([78], p. 1)

This radical transformation that is underway clearly will strain the military as it is fiscally, technologically and doctrine constrained. The tools to understand how decisions will affect the implementation of technologies are difficult, but of even more critical importance is the adaptation in the cognitive domain, where the challenges of relinquishing the grasp of antiquated notions is most critical. President G.W. Bush stated that "transformation requires more than high-tech weapons–it requires creativity, ingenuity, and a willingness to try new things. All the advanced technology in the world will not transform our military if we do not transform our thinking." The tools to assist in performing this leap have been introduced, but no formal process for doing this has been formulated in a structured and tested manner.

Current forces need to be adapted to produce desired effects, but how can we do that in a complex environment? For example, how can we maximize the capability of a family-of-systems for minimum cost? How do we reflect the effects of the complexity of the operations to the design of the technologies? How do we tailor emergent behavior? How do we design the architectures to exploit the correct of set of possible interactions?

The services have been pursuing this new paradigm of warfare. In his U.S. Army transformation monograph, MAJ Mark Calhoun [68] states that "instead of perpetuating the

tendency to establish constricting layers of bureaucracy and control, the Army must engender an environment that keeps the spirit of initiative and innovation alive in peace as well as in war. This can best be accomplished by embracing complexity theory as the intellectual foundation..." This call for the implementation of complexity in the development of military architectures is supported by experts in the field, as for example Moffat [241], who argues that attrition models, fractal models, neural net models, and power spectrum models do not match the historical data as well as Self-Organized Criticality (SOC) models which are based on complexity theory.

CHAPTER II

BACKGROUND

2.1 Reductionism, Holism, and the Evolution of Complexity Science

Complexity Science has received extensive attention in the last few decades. Figures 5 and 6 were created by searching for publications in Google™Scholar that contained the phrases *Complexity*, *Emergent*, *Emergence*, and *Complexity Science* for 2-year periods. The numbers approximately represent, in relative terms, for each 2-year period, how many articles were published that contain these phrases. The graphs clearly exemplify how the field of "Complexity Science" has grown since 1984 and how it seems to continue growing. Note that "Complexity Science" is on a different set of axis, but on a relative growth it is evident that the number of publications containing the phrase is growing exponentially. This can be taken to indicate that "Complexity Science" is an emerging field, and many of the theories, definitions, axioms, etc, are not clearly defined yet. This makes it difficult to work with concepts related to complexity science but it brings an opportunity to innovate and contribute to a field that has not settled yet. The semilog plot (Figure 6) more clearly represents in one set of axes how the number of publications containing the term "Complexity Science" has been increasing relative to the other terms, by demonstrating that the gradient in the logarithmic plot is considerably higher for it than for the others.

In order to understand the emergence of complexity science, it is important to understand the philosophical scientific debate that surrounds it, namely, the debate between Reductionism and Holism. It is not an overstatement that many of the advances in science can be attributed to the 17th century French philosopher, mathematician, scientist and writer René Descartes and his formulation of reductionism. In his "Rules for the Direction of the Mind" (1625-28), he presents a method for "*discovering the truths of nature*" that would eventually lead to the modern scientific method. In part *V* of his *Discourses*,

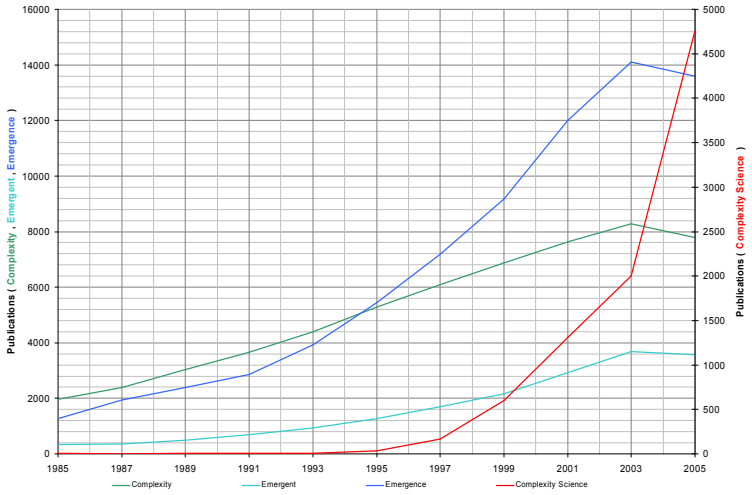

Figure 5: Number of complexity publications as a function of time (In Linear Scale).

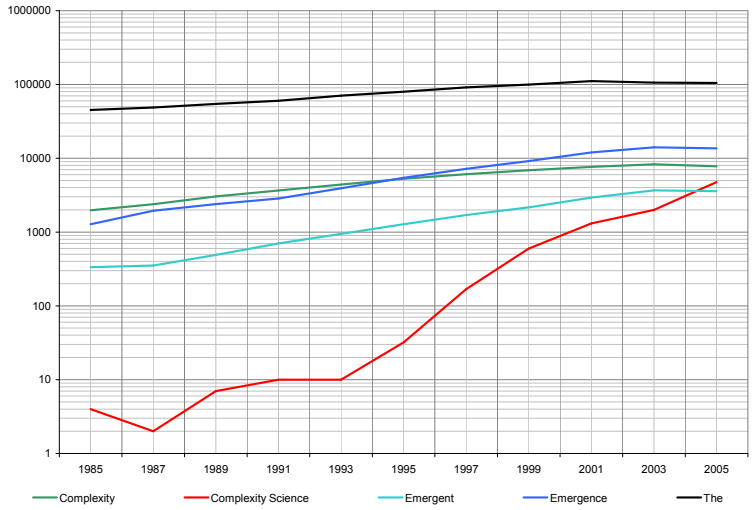

Figure 6: Number of complexity publications as a function of time (In Logarithmic Scale).

Descartes argues that the World is like a machine, and that the machine could be understood by taking its pieces apart. Or in Wilson's words [343], reductionism is "the study of the world as an assemblage of physical parts that can be broken apart and analyzed separately." Reductionism became a vision of reality during the Eighteenth and Nineteenth Centuries, and it is greatly responsible for propelling Western science, e.g., it has helped us move from the air, water, earth and fire notion of antiquity to today's understanding of the structure of the atom [302].

> "No, this trick won't work... How on earth are you ever going to explain in terms of chemistry and physics so important a biological phenomenon as first love?"
>
> - Albert Einstein

In 1926, J.C. Smuts wrote "Holism and Evolution" [304] in which he argued that some systems cannot be studied through reductionism, and can only be studied from a holistic perspective. In a way, Holism, or the "theory of the whole" as Smuts defined it, takes the opposite view from Reductionism, that, as stated by Aristotle in his Metaphysics, the "whole is more than the sum of its parts." This is assumed to mean that the parts of any whole cannot be understood except in relation to the whole, or if we assume that Aristotle meant 'sum' as simply an arithmetic sum, that the parts interact.[1] An argument often posed against reductionism is that even though the laws that describe the behavior of atoms are fairly well known, the behavior of larger systems, e.g., an animal or an ecological system, cannot be described with the same degree of certainty. The focus of Smuts' book is on natural systems and not on engineered systems, and until recently his ideas had not proved to be as suitable of a paradigm for the *hard sciences* (physics, chemistry, etc.), as they had been for sociology, psychology, economics, ecology, and even theology to name a few. Ever since the publication of "Holism and Evolution" the philosophical debate has raged with philosophers from both camps arguing for the validity of one over the other.

[1] For more on this, refer to Peter Schuster's excellent article ([291], p. 11) on the subject.

For the physical sciences and engineering, the development of holism has at the very least helped us redefine our understanding of reductionism, prompting its categorization.

Dawkins [95] argues that it is not fair to argue that reductionism is not a valid approach because we cannot explain the behavior of an ecological system from the behavior of the atom. He terms this extreme form of reductionism "greedy reductionism." He instead proposes the concept of hierarchical reductionism, where it is not necessary to attempt to explain the behavior of one system at a given level from the very lowest level, but only from the level immediately below. E.g., it is not necessary, nor correct, to explain the behavior of an ecosystem from the level of atoms, when it is possible to explain it from the level of animals and their biology. Yet, others argue that even though hierarchical reductionism is a better alternative to greedy reductionism, it still does not address the issues that arise in the study of a complex system.[30] For example, Anderson ([30], p. 393) states that the "constructivist hypothesis breaks down when confronted with scale and complexity". The constructivist hypothesis is often assumed to spawn from the reductionist hypothesis, but in reality the latter does not imply the former, i.e., the ability to break down things to fundamental laws does not mean that everything can be built from said laws. Einstein's previous quote, eloquently proposes this same opinion. This view is supported by Koch and Laurent [198] who state: "Continued reductionism and atomization will probably not, on its own, lead to fundamental understanding." The fact that Koch and Laurent link reductionism, holism, and understanding, is of particular interest to this body of work.

It is important to note that not understanding the parts leaves us with little to work with, and it is necessary if one is to go beyond pure narrative descriptions.([291], p. 11) John Maynard Smith [231] illustrates the case from a pragmatic point of view: "As it happens, I do not understand how modern sewing-machines work, but this does not lead me to suppose that the laws of topology have been broken: Indeed, I feel confident I could find out if someone would let me take one into pieces." This comment was aimed at those that believed that vital forces explained phenomena that could not be explained with the understanding of the time. In the early 19th century, some postulated that the patterns formed by schools of fish or flocks of birds was caused by some higher level force

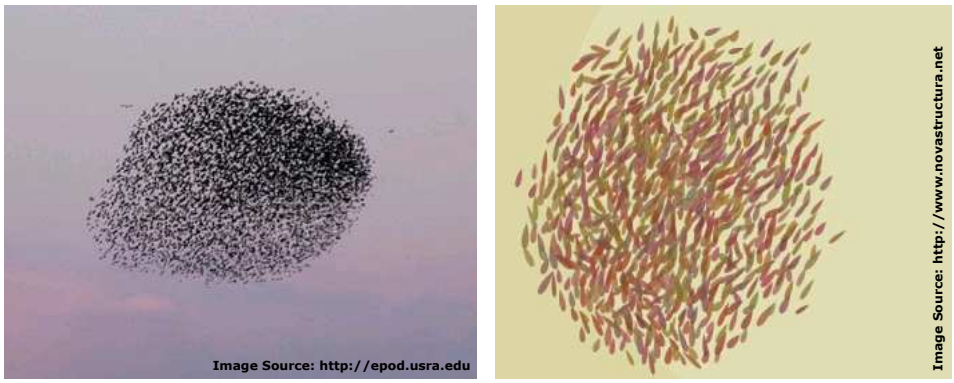

Figure 7: Pattern comparison between flock of real birds and simulated Boids.

or intelligence, when in reality it was demonstrated that very simple rules governing the behavior of each entity produced organized patterns through their interactions. Figure 7 depicts how these highly organized patterns can be recreated with simple rules through their interactions, the figure on the left is a picture of real birds, and on the right is a screen capture of a Boids simulator. This has propelled some to argue that *Complexity Science* should be the "study of the behavior of collections of simple (and typically nonlinearly) interacting parts...".([171], p. 14) This statement proves to be vague when confronted with questions like: "what is a simple part?" and "how big is a collection?"

Ludwig Von Bertalanffy, one of the forefathers of General Systems Theory (GST), stated in his seminal paper "An Outline of General Systems Theory" [332] that the consequences of the mechanistic world-view have been fatal to our civilization and he proposed GST as a way to study the whole. GST is one of many attempts to study the intrinsic characteristics of complex systems. Other techniques have flourished since the mid-20th century, the most notable and influential of these have been J. W. Forrester's System Dynamics and Systems Thinking [132, 131], Maturana's and Varela's concept of Autopoeisis [329], and John Holland's Complex Adaptive Systems [168, 169]. Figure 8 is a timeline of the evolution of techniques that are associated to complexity science along with specific developments that influenced the research associated. The ground-breaking ideas of these researchers have propelled others to dig deeper into the common problems of complexity, but without a doubt, the single most critical enabler in the development of complexity science has been

the exponential growth in computational power.

2.2 Complexity

Going back to the roots of complexity, Warren Weaver in his 1948 article "Science and Complexity" [338] is the first to recognize complexity as a new branch of science. Weaver wrote this article as part of series that was meant to seal his legacy as a prominent American scientist and the Director of the Science Division of the Rockefeller Foundation. As presented by Johnson [182], Weaver goes on to argue that scientific inquiry during the last few centuries can be divided into three camps. First, the study of *simple* systems, those consisting of a few variables, e.g., planetary motion, calculation of voltage from a resistance and a current, etc. Second, is the camp of *disorganized complexity*, problems of millions or billions of variables that can only be approached by the methods of statistical mechanics and probability theory, e.g., Boltzmann's work. The third, and final, camp identified by Weaver was the middle region, those problems between the very few and the very many variables. Of this region he wrote:

> The really important characteristic of the problems of this middle region, which science has as yet little explored or conquered, lies in the fact that these problems, as contrasted with the disorganized situations with which statistics can cope, show the essential feature of organization. In fact, one can refer to this group of problems as those of *organized complexity*... [emphasis added]
>
> The importance of this middle region, moreover, does not depend primarily on the fact that the number of variables involved is moderate–large compared to two, but small compared to the number of atoms in a pinch of salt. The problems in this middle region, in fact, will often involve a considerable number of variables. The really important characteristic of the problems of this middle region, which science has as yet little explored or conquered, lies in the fact that these problems, as contrasted with the disorganized situations with which statistics can cope, show the essential feature of organization. In fact, one can refer to this group of problems as those of organized complexity.

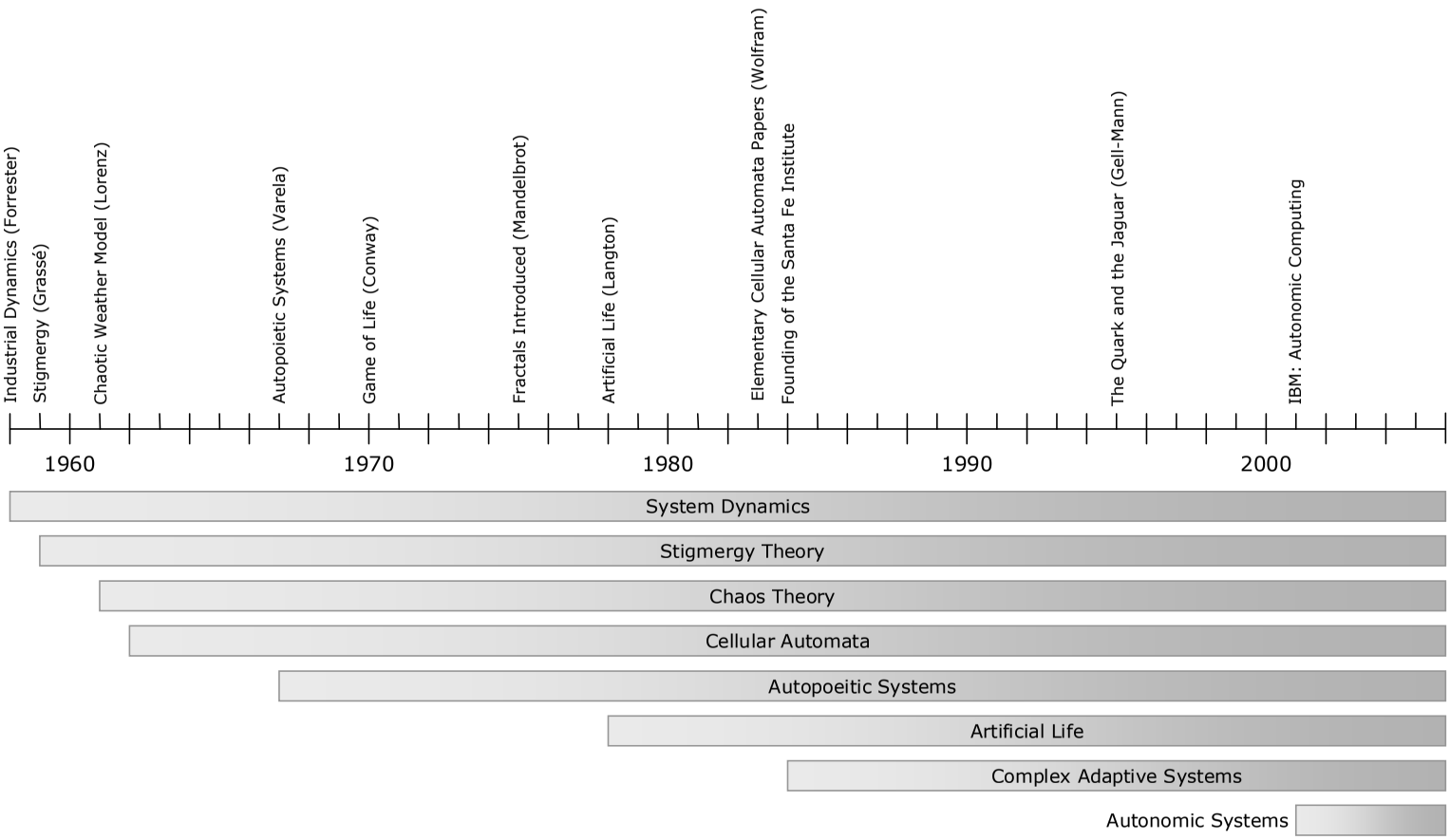

Figure 8: Mayor events in the evolution of complexity science since 1958.

> Much more important than the mere number of variables is the fact that these variables are all interrelated... These problems, as contrasted with disorganized situations with which statistics can cope, show the essential feature of organization. We will therefore refer to this group of problems as those of *organized complexity*.
>
> - Warren Weaver [338]

Weaver may have been the first to recognize the emerging scientific field of *organized complexity* but he failed to provide a strict definition for complexity. He did mention that the variables are all interrelated, which as it will be made apparently shortly, is a key characteristic of complex systems. In Weaver's defense, researchers to this day struggle to agree and reach consensus on the definition of what is a *complex system*. For an eloquent article on the subject, the reader is referred to Peter Corning's article [86] "Complexity is Just a Word!". Corning argues that despite the extensive efforts to define complexity, it is still in the eye of the beholder. He quotes Seth Lloyd, a leader in the field of complexity science, as saying, that when asked to define complexity he could not do it, but he could recognize it when he saw it. In his article, Corning finally argues that what most researchers and thinkers have been able to do is enumerate a series of characteristics exhibited by complex systems. He proposes three characteristics of complex systems: (1) they are composed of many parts, (2) they are highly interdependent, and (3) the parts produce synergies (combined effects) that are not easy to predict from the individual parts by studying them in isolation. Simon's presentation for the 1976 Biennial Meeting of the Philosophy of Science Association [297] contains the first extensive enumeration of characteristics of complex systems. He lists the following characteristics as those generally related to complex systems: (1) large number of components, (2) systems that exhibit more interdependence between the components, (3) systems that are *undecidable*, and (4) heterogeneous, meaning that the components are different. This is a physical, or organizational, characterization of complex systems. These characteristics are still regarded as intrinsic to complex systems. The term undecidable system means that there is no effective method for determining whether any

given well-formed formula is a theorem, i.e., when a statement is made about the system, there is no simple procedure that can be followed to determine whether the statement is true or false.

Murray Gell-Mann, a Nobel Prize winner who is a proponent of complexity science and co-founder of the Santa Fe Institute, wrote a paper in 1995 [147] attempting to answer the question: "What is Complexity?" In his brief treatise he states that many quantities have been proposed to measure complexity and that many are needed to capture our intuitive ideas of what is meant by complex. Initially, he concentrates on traditional measures of complexity (e.g., computational complexity, algorithmic information content, etc.) to finally delve into what does complexity entail for someone interested in the behavior of the universe. He proposes that the complexity of an entity can be best described as: "the length of a concise description of a set of the entity's regularities." This definition presents a critical problem in that there is no procedure for finding all regularities of an entity. Gell-Mann's paper goes further than defining complexity, he attempts to quantify it, but it is not clear that complex systems were defined in the first place.

Dozens of other papers and books have been written where a definition of complexity is at least attempted [31, 40, 56, 74, 86, 87, 88, 96, 119, 146, 168, 171, 221, 241, 250, 272, 297, 317]. Most of these concentrate on understanding the characteristics of complex systems. It seems apparent that every author proposes a slightly different set of characteristics for what is commonly accepted as complex. An extensive list of characteristics with a brief description has been compiled below:

Dynamism The complexities of systems only arise as time progresses, if the systems are frozen in time, they do not have an opportunity to behave in a complex manner. This is generally implied in all the definitions presented, but it is clear that what is complex about complex systems, is their behavior, which exists only in time.

Number of Elements The system is composed of a large number of parts as recognized by Corning ([86], p. 199), Simon ([297], p. 1), and others.

Interdependency Corning, Simon, and others recognize that complex systems exhibit

significant interdependence between the parts, often having time-dependent interdependencies, with relations appearing and disappearing over time.

Nonlinearity The effects due to interactions are not proportional to the causes, i.e., small inputs produce large outputs, and viceversa. E.g., if-then rules (micro-nonlinearity), feedback cycles (macro-nonlinearity), etc. [171, 213]

Irreducibility The complex system loses macro-behavior if system is broken into its elemental pieces or if portions are removed ([172], p. 3).

Hierarchies The elements that compose the complex system can be grouped in multiple scales in space and time, e.g., agents aggregate to form a system that can be described as an agent, which in turn can aggregate with other super-agents to form super-super-agents ([171], p. 10).

Emergent/self-organizing behavior No central control or plan ([56], p.3); "more is different"; reductionism does not imply constructivism ([30] p.93)

Many nearly degenerate/equivalent configurations This means that the system can have the same macro-state under different micro-states. For engineered complex systems, this means that if the system is designed from requirements (top-down) non-unique solutions are possible.

Adaptation Ability to respond to changes in environment; explicit or emergent; relation to stability [168].

Biological/"life-like" behavior Self-reproducing, adaptive, evolving, learning [54]

"Intelligent agents" Individual parts of systems have internal schema; e.g., "if/then" rules. (N.B.: Intelligent agents are neither necessary nor sufficient for intelligent behavior of the whole system: cf. the brain and the Congress.)

Non-equilibrium Order The system is seldom in an equilibrium state ([172], p. 3).

Collectivist Dynamics There is a continual feedback between the behavior of low-level and the high-level ([172], p. 3).

These characteristics can be grouped into two categories, those that are evident at the elemental (micro) level of the system, and those that are observed at the macro-level as shown in Figure 9, and as with nonlinearity, sometimes both. The elemental basic required property that all complex systems must have is a large number of elements, a sizable number of relations (interdependence) and set of shared rules by which they operate. These basic features can give rise to a larger set of properties at the micro-level of the system. These are not required, or exclusive, to give rise to a complex system, but are generally present in complex systems. At the macro-level we also can categorize two sets of characteristics, the observable phenomena and the features, or properties. The phenomena are related to the emergent capability of complex systems, allowing them to produce patterns at the macro-level that cannot be induced from simply analyzing the parts in isolation. Examples of these are, self-control, self-organization, self-replication, etc. These are the behaviors that are often related to the term "self-organized criticality," a concept that says that complex systems have a tendency to exist at the "edge of chaos," a region between stability and chaos. The large number of attempts to define complexity has left researchers in a quandary, every source seems to attempt to define these behaviors from a different angle in the hopes of capturing the *intuitive* concept of complexity. Some have argued that complexity has become a *buzzword science* ([171], p. 3) implying that there is more emphasis placed on the buzzwords than on the science behind it [86, 74]. Complexity literature is littered with terms like emergent [96, 106, 242, 173, 310, 181, 36], adaptive [28, 51, 210, 68, 74], edge-of-chaos [31, 238], and self-organized criticality [181, 228, 74], yet consensus on a strict definition for these does not exist, nor is there a clear and accepted relationship between these and complexity.

This seemingly endless debate on the characterization of complexity has propelled some researchers to take unconventional paths in defining complexity. An original example is the proposed characterization put forward by Professor Nam P. Suh [315] of MIT, whereas others analyze the physical characteristics of complex systems, Suh argues that complexity must be defined in the functional domain. He stipulates that systems whose functionality deviates from the specified range behave in a complex manner, and those systems that

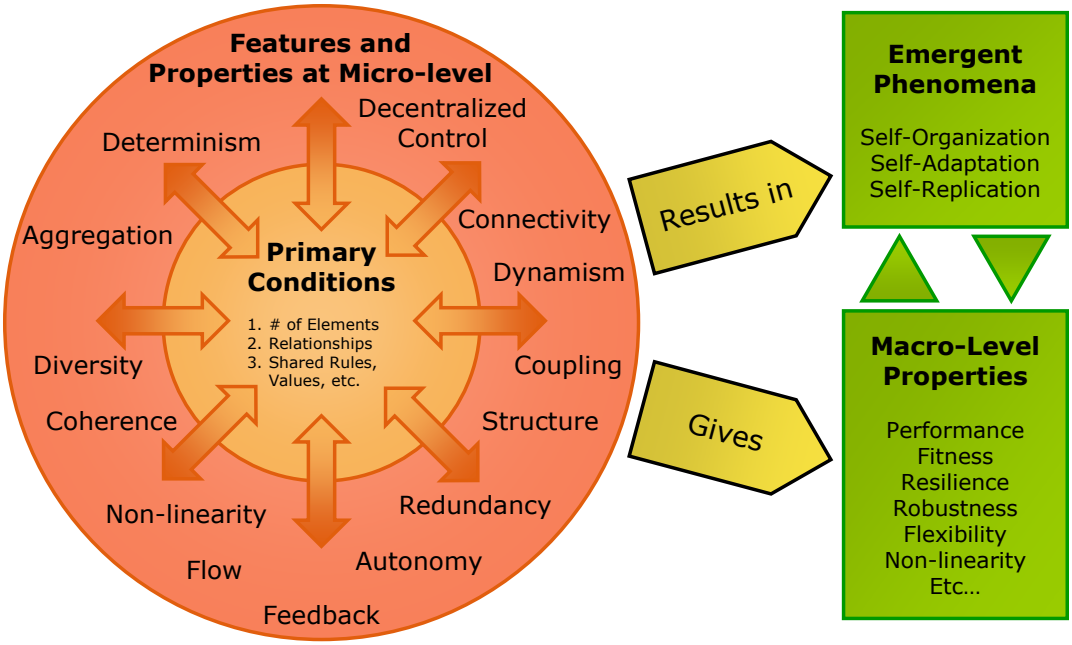

Figure 9: Features and Properties of Complex Systems ([89], p. 53).

regardless of how complex they are in the physical domain, can be considered to be simple if they perform their functions within the required ranges. An important note on Suh's definition is that it is suitable to engineered systems, and less useful to natural systems, such as bee colony or a socio-cultural system. In his book, Suh identifies four types of functional complexity, he refers to these as: (1) (time independent) real complexity, (2) (time independent) imaginary complexity, (3) (time dependent) combinatorial complexity, and (4) (time dependent) periodic complexity. Real complexity is intrinsic to the implementation of the system because the uncertainty in the functionality of the system cannot be reduced to lie within the desired range. Imaginary complexity arises from lack of understanding of the system design, architecture, and/or system behavior; it is a functional complexity that does not exist in reality, only as a byproduct of the relationship between the user and the system. Combinatorial complexity arises when the system functional range drifts over time and therefore cannot be predicted. Periodic complexity is similar to combinatorial complexity, except it has the characteristic that the system follows a repeating pattern

(this does not have to be a temporal pattern necessarily, it can be geometric, thermal, electrical, chemical, etc). Suh's most important recommendations relate to transforming a system, or architecture, from one type of complexity to another with the goal of reducing functional complexity. He argues that combinatorial complexity should be reduced by transforming it to periodic complexity, that real complexity should be reduced by selecting uncoupled functional requirements and that imaginary complexity can be reduced to zero if the functional requirements are completely independent. This view of complexity from the functional domain provides some insights into what complexity means, but it does not assist with the task of understanding complex systems.

Up to this point we have identified two camps attempting to define complexity. On the one hand are the people attempting to characterize it based on what the systems exhibit in the physical domain and on the other is the characterization based on the functional domain. The definition proposed here is based on a synthesis between these two and is depicted in Figure 10. If a system is simple in both physical and functional domain, it is safe to characterize it as a "simple" system. These systems behave within their required tolerances and are composed of a few elements, or independent elements that interact fairly linearly. On the other extreme there are the systems that are both complex in the functional and physical domains. These systems are composed of many entities, that are highly interdependent and interact nonlinearly, while at the same exhibiting a larger-than-acceptable degree of uncertainty in their ability to perform the required functions. The two other alternatives are the systems that are either complex in the physical or functional domains, and viceversa in the other domain. What can we say about these systems? For those that are complex in the functional domain, but simple in the physical domain, traditional systems engineering provides sufficient tools to achieve the desired results by redefining the functional requirements and redesigning the system and the processes involved. Of the final quadrant of this complex system, of those systems that are functionally *simple* but physically complex, what is to be done is not clear. Suh argues that as engineers our work is done, the system performs its function within the prescribed ranges and is therefore suitable for achieving the required tasks. The question then becomes, how can we ensure that our system will behave

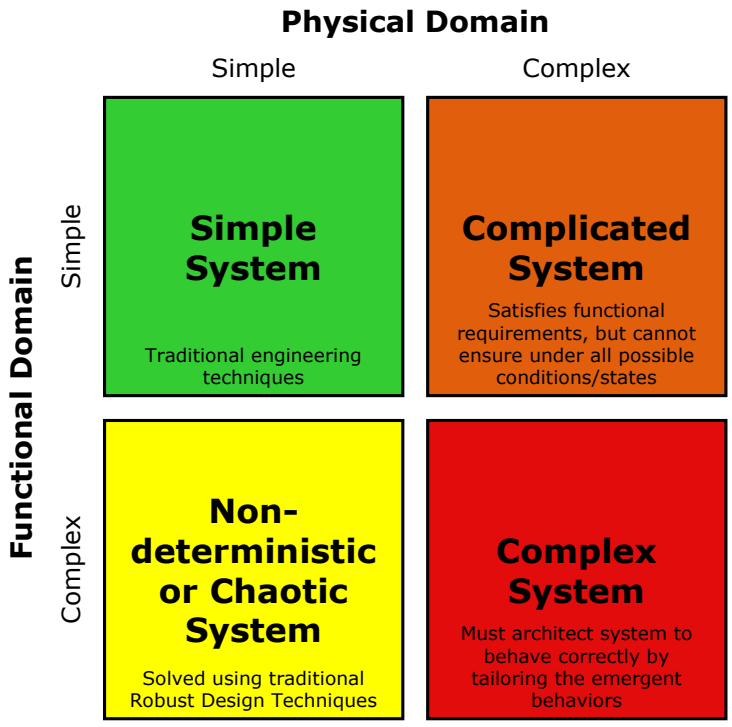

Figure 10: Complexity Matrix.

in this manner under all possible situations since it is physically complex and all possible states of the system cannot be tested or known a priori? This reason supports the idea that physically complex systems lead to complexity, while functionally complex systems that are physically simple, can be tailored to behave in a simple manner. To reinforce this idea, the examples that Suh utilizes in his book as combinatorially complex and periodically complex, are also physically complex.

In the 1940s John Von Neumann, working from a recommendation from his friend Stanislaw Ulam, developed a simple artifact to study the requirements for self-replicating machines. His development became the first cellular automaton. Cellular automata are composed of a grid of cells with finite dimensions, each of which can only have a finite set of states. Every cell also shares the same rule for updating its state based on the state of its immediate neighbors. The link between cellular automata and complexity is that very complex patters can be created with simple rules. A good example of this is rule 110 cellular automata, a simple set of rules of one dimensional automata that produces a pattern that

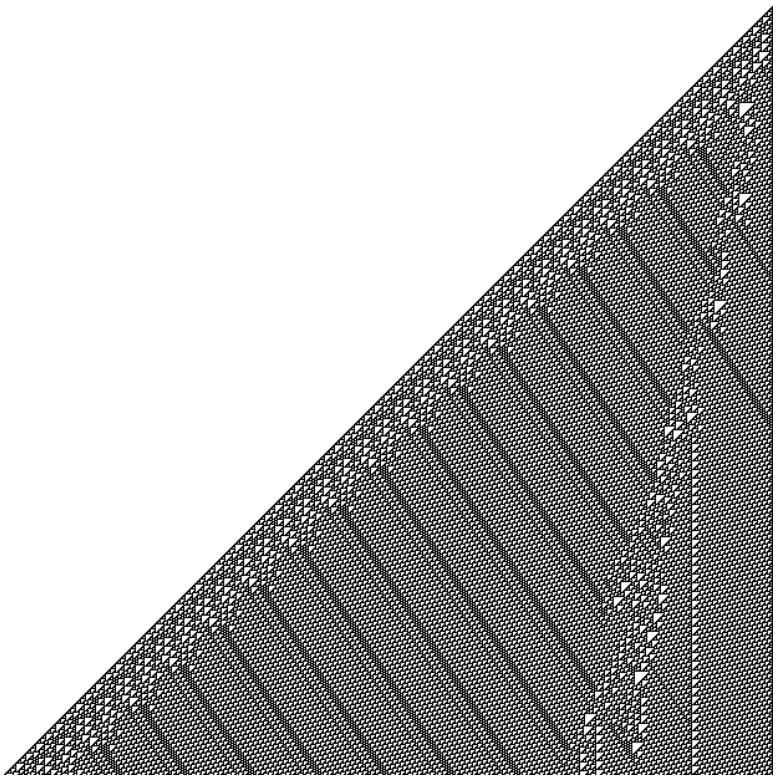

Figure 11: Rule 110 cellular automata with random initial conditions [347].

is neither chaotic, cyclic, yet it never reaches steady state. Figure 11 depicts a pattern produced by rule 110 automata, the horizontal axis is the 1-D cellular automata, and the vertical axis is time advancing downwards. Note that the patterns do not repeat themselves, yet they are not random. Wolfram [345, 346] was one of the first to propose Cellular Automata as a generic model for complexity to study how complex behaviors/patterns can arise from simple rules. In his new book, *A New Kind of Science* [347], Wolfram argues that instead of using mathematical equations to study nature, scientists should pursue the development of simple programs that can reproduce the complexities that natural systems exhibit. Wolfram believes that this new study of nature has the potential to revolutionize science by shocking its very foundations.

Adam M. Gadomski, a nuclear physicist turned socio-cognitive engineer, proposed an interesting property that is common to complex systems, he states that "the common property of complex systems is the difficulty of their formal modeling." Bankes ([39], p. 7263)

goes further than Gadomski in asserting that "any system whose behavior is well captured by some model cannot be complex, under most definitions of complexity." This is of particular interest to this work in that it relates the objects of interest (complex systems) with the goal (to model them). If the common property of complex systems is that they are difficult to model, and modeling is required to understand complex systems, it is natural to say that complex systems are those that are difficult to understand. This brings us full-circle with the definitions proposed previously by Lloyd, Corning, and to some degree Weaver himself. Is this difficulty to understand rooted in the nature of "complex systems" and the human brain, or is it that a lack of mathematical formality for them does not allow us to "wrap our minds" around complexity? This question dwells deep into the philosophy of cognition and science. For example, were the behaviors of the planetary system "complex" to the minds of the ancient astronomers before Copernicus, Galileo, Kepler and Newton came along? The models they developed were certainly complicated at first, and became simpler as they were understood better. Where the behaviors of gases "complex" before Boltzmann developed the statistical techniques used to describe their macroscopic relations? Is "complexity" as we know it an artifact of our inability to develop "simple" models to describe the behaviors of these systems? The answer is that diverse fields as biology, sociology, engineering, policy, psychology, and ecology to name a few, have encountered problems that share many features with one another. Mathematics has been attempting to develop tools to study these problems, but a widely acceptable set of tools has not been agreed upon.

Going further, Steven Bankes [39] argues that it is not possible to optimize complex systems and introduces the concept of "deep uncertainty," which he defines to be "the result of pragmatic limitations in our ability to use the representational formalisms of statistical decision theory to express all that we know about complex adaptive systems and their associated policy problems." His argument relies on the fact that the intrinsic assumptions supporting probability theory are not applicable to complex systems. Cares ([72], p. 16) supports this belief when he states that "when assessing complex systems, it is not possible to know the odds, regardless of the amount of information," he recommends that "it is

better to understand a systems dynamics" instead. This understanding is what will enable designers to develop systems that produce the required emergent behaviors, and will empower operators to make the most appropriate decisions, in short, develop *heuristics* to help you make decisions.

2.3 Modeling

> "All models are wrong, some models are useful."
>
> - George E.P. Box

This truism of Operations Research (OR) is key to the efforts of this body of work. It elegantly and succinctly implies what a model is and what is its purpose, namely a useful approximation. The use of models, in its most generic sense, is something that we do as humans on a constant basis. Plato wrote of the Forms (ideal patterns) which we recognize in reality, even though they are idealizations (models) of reality, that allow us to comprehend reality.[184] The subject of modeling is as broad as human intellect itself since it is its tool of choice.[281] The discussions in the literature range from abstract to specific, and this wide range of interpretation is the reason why there are so many definitions, interpretations, and classifications.

Continuing with the definitions, a model can be defined as a "A simplified or idealized description or conception of a particular system, situation, or process, often in mathematical terms, that is put forward as a basis for theoretical or empirical understanding, or for calculations, predictions, etc.; a conceptual or mental representation of something."[4] This definition implies that models are mainly used for two purposes, (1) for aiding in the understanding of a system's behavior, or (2) for aiding in predicting its behavior. The DoD defines a model as "a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process" ([98], p. 138). That same document defines the act of modeling as the "application of a standard, rigorous, structured methodology to create and validate a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process" ([98], p. 138). Modeling and Simulation (M&S) is defined as "the use of models, including emulators, prototypes, simulators, and stimulators,

either statically or over time, to develop data as a basis for making managerial or technical decisions." The terms "modeling" and "simulation" are often used interchangeably ([98], p. 138), but this last definition seems to be in contradiction to the DoD's definition of simulation: "A method for implementing a model over time" ([98], p. 157) which indicates that simulation only takes place over time, while previously, it was stated that simulation can be done independent of time, e.g., by running a Monte Carlo Simulation (MCS) over the parameters of the model and obtain distributions of responses. For the purpose of this thesis, a model will be described as a representation of something for the purpose of aiding in understanding its behavior, or making predictions of its future states. Modeling will be defined as the act of creating models, from their conceptualization, development, and if necessary, testing. Simulation will be defined as the exercise of a model to obtain insight into its behavior (either dynamically or statistically). Finally, M&S will be defined as the additive process of modeling and of simulation, i.e., the process of conceptualizing, developing, and if necessary testing the model, followed by the exercise of that model to study its behavior. This process may be iterative, with the model being adapted as a result of its simulation's output.

In summary, for the purposes of this thesis, these will be the definition for the following terms:

Model An abstract representation of a system developed to aid in the understanding and/or predicting of its behavior.

Modeling A rigorous method for creating and testing models.

Simulation The exercise–either statistically or over time–of a model.

Modeling and Simulation The additive process of modeling and of simulation; the rigorous process of conceptualizing, developing, and if necessary testing the model, followed by the exercise of that model to study its behavior.

2.3.1 The purposes of Modeling and Simulation

One of the definitions presented previously implied that modeling can be used for two purposes, (1) aiding in understanding, and (2) predicting future states. The literature implies that modeling is generally done for one or more of three possible purposes: (1) Exploration, (2) Explanation, or (3) Extrapolation. **Exploration** is the process of searching for the purpose of discovery. **Explanation** is the process of discovering causalities, i.e., linking causes to effects, and therefore increase understanding. **Extrapolation** is the process of predicting outside of the known and observed, this is by far the most treacherous use of modeling and the one that should be done with the most care. These three distinctive purposes have different requirements on models, and when created for one purpose and used for another, they can be more or less dangerous based on the degree of misappropriation and misuse.[170] Models used for exploration are validated within a range, and tested for their generality within that range. They are effectively interpolating between observations, and are therefore the most benign. Explanation tends to employ models that share the same characteristic, but can help elucidate causation, if not simple correlation. Ensuring that correlation is not confused with causation, is an important step, their use tends to require a higher degree of skill from the analyst and modeler. Extrapolation, as it has been stated before, is the more risky of the three ventures, requiring the most care. Results from these extrapolations should be presented with a disclaimer stating that the results are by no means to be taken as fact. Furthermore, it is not appropriate, some argue ethical [220], in most cases for these extrapolations to be done in a deterministic manner, since there is an inherent degree of uncertainty associated with them. The manner as to how this uncertainty should be captured is not evident in many cases, and may require extensive understanding of the supporting theory (e.g., probability, possibility, statistics, etc.) and knowledge about the problem.

A similar but slightly different understanding of the purpose of models is the one used in the system analysis and decision support communities.[281] They tend to use models to gain control over reality, to make decisions, or answer questions about the world. These communities tend to use models for two purposes, for describing or for prescribing. **Describing**

involves explaining reality, this is the same as **Explanation**. **Prescribing** involves identifying optimal solutions to problems, which in a way is the synthesis of exploration and extrapolation, since to identify an optimal solution, potential solutions must be explored, and if non exist, extrapolations to previously unchartered solutions must be pursued. In the literature, prescriptive models are often subdivided into those used for normative uses (i.e., to identify feasible goals and standards) and idealization (i.e., construction of hypothetical, ideal entities that "illuminate real-world phenomena" [281]). Rothenberg further argues that specific uses of models include forecasting with conditions (projection), forecasting without conditions (prediction), allocation and derivation, hypothesis testing, experimentation, and explanation. These once again, can be classified as special cases, or synonymous, with with three purposes specified initially.

2.3.2 Taxonomies of Modeling and Simulation

With clear definitions for each of the terms of interest related to M&S, and the purposes for which it is done identified, the next step is to study the existing methods and techniques. The number of techniques and methods is too large to enumerate and analyze in a useful and comprehensive manner. A more organized approach is to classify them and study the different classes. With a proper taxonomy and categorization of modeling techniques, it may be possible to identified the most useful to the needs of understanding complex systems, and if non exists, understand which combinations of characteristics are not covered, and bridge these by developing new or hybrid techniques.

An extensive literature search yielded no universally accepted taxonomy[2] (nor ontology[3] for that matter) of modeling and simulation techniques. Several have been proposed, but in general they are too vague for them to be useful–as will be explained–or too specific to be applied on a broad spectrum of modeling and simulation techniques–e.g., [52, 128, 129, 130]. The problem of creating a taxonomy for models can go ad infinitum since as explained before, models can be extremely generic in nature, e.g., Blackwell and Engelhardt [55] proposed a taxonomy of diagram taxonomies, which contain from a cognitive science and

[2]The arrangement of entities within a set based on the characteristics that define them.
[3]The characteristics of a set of entities that are used to classify and distinguish them.

linguistic perspective, a thorough review of the different taxonomies related to–what eventually amounts to–modeling. From the engineering perspective, Przemieniecki [270] and Dieter [105]–as many authors in the literature do–propose the classification presented in Figure 12. In their classification, models can belong to one of three categories, (1) iconic, (2) analog, or (3) symbolic. Some authors argue that there is a fourth basic type of model, a (4) logic model, although this can be considered a subcategory of symbolic models. **Iconic models** are visual representations of what real "things" look like, e.g., maps, scale models of aircraft, etc. They represent entities rather than behaviors. **Analog models** can be considered the antithesis because they focus on representing the behavior of a real thing while not necessarily being similar to the real entity of interest. Analogies in engineering have served an important role, but as engineers became more empowered, and therefore dependent, on computers their use has declined.[4] **Symbolic models** nowadays the most used, and arguably the most useful models in engineering, are those that, as described by Dieter ([105] pp. 249), abstract the "important quantifiable components of a physical system." Dieter proposes dividing this category into theoretical and empirical, i.e., those based on "established and universally accepted laws of nature" and those that are "the best approximate mathematical representations based on experimental data." Przemieniecki on the other hand proposes the categorization of symbolic models into the descriptive and mathematical, the former representing reality by using words or diagrams, and the latter by quantitative or logical relationships. The mathematical models can be subdivided amongst analytical and simulation models. These two categories describe how the model can be solved or operated on. Analytical models determine "exact quantitative numbers representing the reality," whereas simulation models cannot be solved explicitly, and must therefore be solved through simulation, these tend to involve problems with complex relations, uncertainty, and risk. Przemieniecki further subdivides simulation models into two categories, deterministic and stochastic. This last category, that of discriminating between deterministic and stochastic models, has been used extensively in the literature. In fact,

[4]For an interesting treatise on the use of analogies in engineering, the reader is referred to Murphy et al. [247].

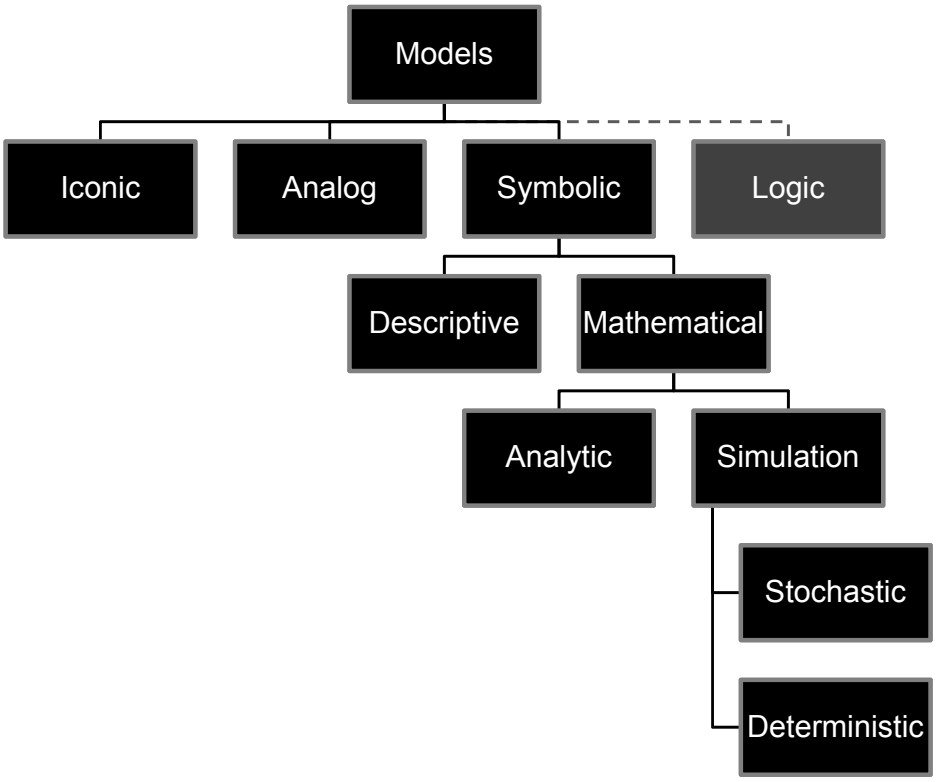

Figure 12: Model Classification ([270], pp. 320)

the most commonly cited taxonomy for models is by the nature of the technique, these are taxonomies of models based on whether they are (1) stochastic or deterministic, (2) static or dynamic, and if dynamic, whether they are (3) continuous or discrete.

Stochastic models attempt to capture uncertainty by providing outputs that are not a single value, but a distribution of possible outcomes. Therefore, their results are only valid within their statistical significance, which means that, in some cases, a large number of repetitions need to be executed to obtain it. **Deterministic models** are easier to implement, and considerably cheaper to execute, but seldom their results are useful for supporting decisions since they provide a very small slide of the possibilities that may take place. Expected-value models are the most common deterministic models, but they carry the assumption that the average is all that the decision maker is concerned with, when in

reality he/she may be interested in a robust solution, in which case the average may not be meaningful to base a decision on. Lucas [220] offers an in-depth analysis of when the average will not be sufficient to make the right decisions.

Static models do not describe behaviors over time, they only represent time-invariant results or effects. **Dynamic models** on the other hand model the state of a system over time. They may model it at the macro-state, or aggregate level (e.g., system dynamics), or at the micro-state, or entity-level (e.g., agent-based modeling, discrete event simulations). Additionally, dynamic models can be of two kinds, **continuous** (where change takes place in infinitesimal steps) or **discrete** (where change takes place at distinct points in time). Models that contain both continuous dynamic representations–e.g., $\frac{dy}{dt} = ry^2$–and discrete event representations–e.g., at time $t = 5$ make $y = 15$–are referred to as mixed models, or hybrid models. Discrete dynamic models can in turn be evolved using two techniques–this offers an additional sub-taxonomy–**time-stepping** or **event-driven**. Time-stepping consists in evolving the model over time by the same amount of time, e.g., for a time-step of 5, the model would be evolved from $t = 0$ to $t = 5$, to $t = 10$, etc. Event-driven consists in skipping time to the moment when an event happens. This is useful for stochastic processes that involve queuing, where the event of happening is derived from a stochastic distribution and is therefore a computationally inexpensive process. Buss and Sánchez have nonetheless proposed a discrete-event approach to modeling movement and sensing, where the model is evolved to the moment the next agent-to-agent interaction occurs, in this case, when an agent falls within another's detection radius.

This taxonomy in matrix form, as proposed by Gustafsson [157], is presented in Table 1. Regardless of its veracity, this simple taxonomy does not provide sufficient insight to be useful in ensuring that every aspect of simulation has been covered by the existing techniques. Many dimensions of modeling have not been addressed in this two-category decomposition, e.g., ability to capture entity decision making, to represent hierarchies, to capture nonlinearities, etc.

Table 1: Classification of Modeling and Simulation Techniques. [157]

	Static	**Dynamic**
Stochastic	Statistical Models (e.g., Monte Carlo Simulation)	Actors with Logical Behavior Stochastic Differential and Difference Equation Models (e.g., Discrete Event Simulation and Markov Simulation)
Deterministic	Algebraic Models (e.g., Spreadsheet Models)	System of Differential Equations (e.g., System Dynamics and Dynamical Simulation)

2.3.2.1 Combat Modeling and Simulation and their Taxonomies

The DoD has made extensive use of modeling and simulation since WWII. In the 1960s, with the advent of the computer, the use of models grew exponentially, in their survey (circa 1972) Brewer and Shubik [61] estimate that the investment for all active Models, Simulations, and Games (MSG) in the DoD has been between $130 and $140 million, by 1980 the cost was estimated [135, 325] at $250 million, in 1994 the cost was estimated to be between 1.3 and 1.6 billion dollars, and since then the costs have been estimated to grow geometrically.[256] This major investment in M&S makes it not surprising that the DoD has embraced various modeling and simulation taxonomies. Three DoD taxonomies have been identified. Those based on the model's *representation of reality*, those based on the *degree of aggregation*, and those based on the particular *application* of the model. Unlike the generic taxonomy presented before, the DoD taxonomies are more specific, focused on models and simulations, and not on the techniques used to create them. This makes the taxonomies less applicable to general concerns of M&S, but there is a relation between the modeling and simulation taxonomies and the techniques that can be used for each category.

The first taxonomy and most widely accepted taxonomy embraced by the DoD community is a classification of simulations based on their representation of reality. The DoD M&S Glossary (DoD 5000.59M) in the "Modeling and Simulation Master Plan" (DoD 5000.59P)

specifies three classes of military simulations: (1) Live Simulations, (2) Virtual Simulations, and (3) Constructive Simulations.[98] *Live Simulations* are characterized as "involving real people operating real systems;" *Virtual Simulations* as those "involving real people operating real systems;" and *Constructive Simulations* as those "involve simulated people operating simulated systems."[134] Figure 13 contains the three categories as described by the matrix of real and simulated systems/environment, and decision makers. When forced into this matrix form, the taxonomy clearly shows that a category is missing, the *Artificial Intelligence (AI) testing* or *simulation*. The matrix further exemplifies the problem with this taxonomy, the categories are not discrete nor exclusive. For example, in a flight combat simulator (virtual simulation), there are virtual decision makers embedded as the adversary decision makers, and possibly as friendly superiors (e.g., flight controllers) and subordinates (e.g., wingmen). The *Interactive Simulation* category further demonstrates the inadequacy of a discrete set of simulation classes. In an interactive simulation, the input of the user is required throughout the simulation but the entities in the model are more autonomous than in a virtual simulation. In a constructive simulation, the user input is not required at any moment throughout the simulation. Constructive simulations enjoy the benefit of not requiring any human input, making them less costly–and with some additional effort–repeatable. They are therefore the preferred choice when evaluating a large number of alternatives and/or possible scenarios. As the alternatives of interest decrease and the critical scenarios are identified, virtual and finally live simulations can be conducted. This concept is illustrated in Figure 14, where the notional use of the three simulation approaches is illustrated as a function of the product acquisition phase.

The second DoD taxonomy of models is based on their degree of aggregation. For example, the U.S. Air Force has create a hierarchy[5], presented in Table 2 that contains a general description of what type of model is used for which purpose.[273] The first category, Engineering Analysis, deals with individual systems and components and is focused in understanding the underlying physics governing their behavior. The second category, Weapon

[5]The U.S. Army has a very similar hierarchy based on the resolution of the model. (1) Engineering Models (e.g., munition effects), (2) High-Resolution Models (e.g., weapon-vs-weapon), (3) Medium-Resolution Models (e.g., unit-vs-unit), and (4) Low-Resolution Models (e.g., force-vs-force).[23]

Figure 13: Taxonomy of Military Simulations.

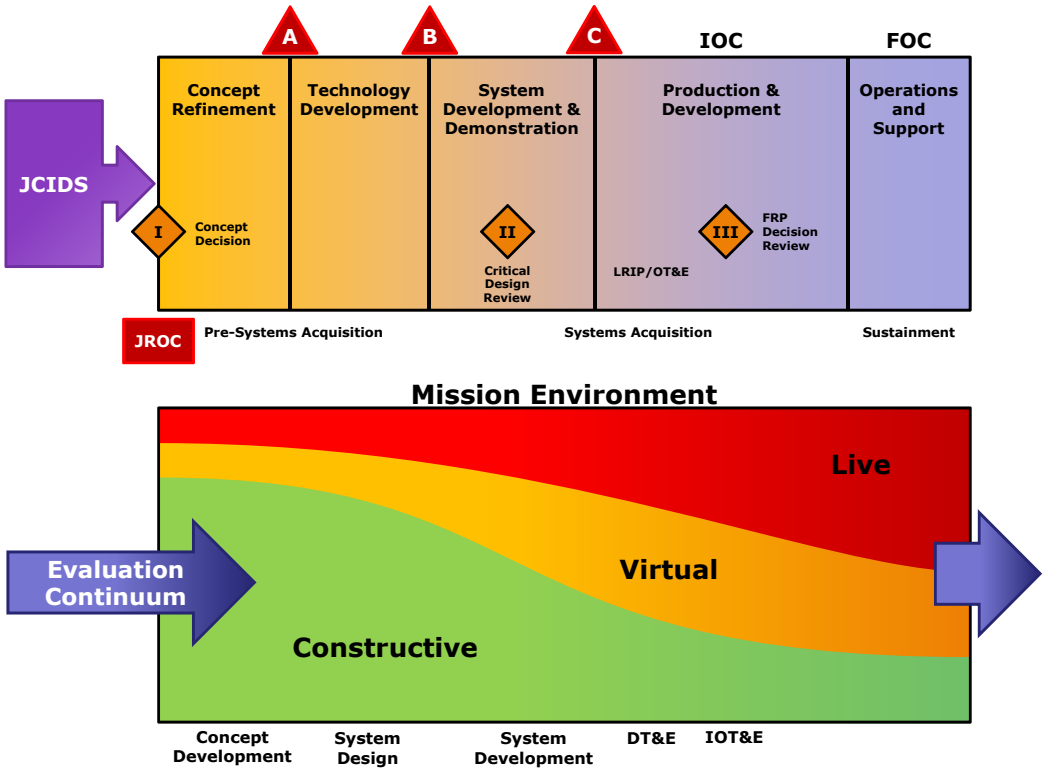

Figure 14: Evaluation Continuum over the Product Life Cycle ([54], pp. 355)

Table 2: U.S. Air Force Hierarchy of Models. ([273], pp. 34)

Level	Simulation Purpose	Simulation Focus
I	Engineering Analysis	Physical Process $n < 50$
II	Weapon System Capability	One-on-one
III	Combat Capability	Many-on-many
IV	Campaign Results	Force-on-force

System Capability, the focus is on how a component affects a platform, e.g., an aircraft and a radar by simulating how well they engage a single enemy aircraft. The third category is concerned with how the platform contributes to the tactics and methods used in a combat mission. It includes considerations for mutual support, command and control, an order of battle and required maneuvers. The fourth and final level, Campaign Results, encompasses all operational activities for a joint campaign (involving all the other branches of the DoD).

The discrete nature of this hierarchy of models is not always representative of the models used by analysts. Figure 15–based on a figure by Soban [305]–describes the continuum of models starting with engineering (also described as phenomenological models [170, 120]) all the way to campaign-level models. In her dissertation, Soban argues that the discrete pyramid is not representative of what exists in the Air Force (and due to their similar use of models, the rest of the armed forces), but that a continuum is more applicable to the current paradigm of M&S in the Armed Forces since the level of granularity is often tailored to particular applications. Davis and Bigelow [92, 93, 53] have been advocating for the use of Multi-Resolution Modeling (MRM) for decision support on issues of military policy, doctrine, and materiel acquisition. In MRM, the models "resolution" (or granularity) can be tailored to the specific study or question of interest, the idea is to use the correct level of abstraction for every question, working at the most aggregate level whenever possible.

The third taxonomy identified in the literature is based on the application of–i.e., the specific functional area addressed by–the model.[23] An example of this taxonomy is (1)

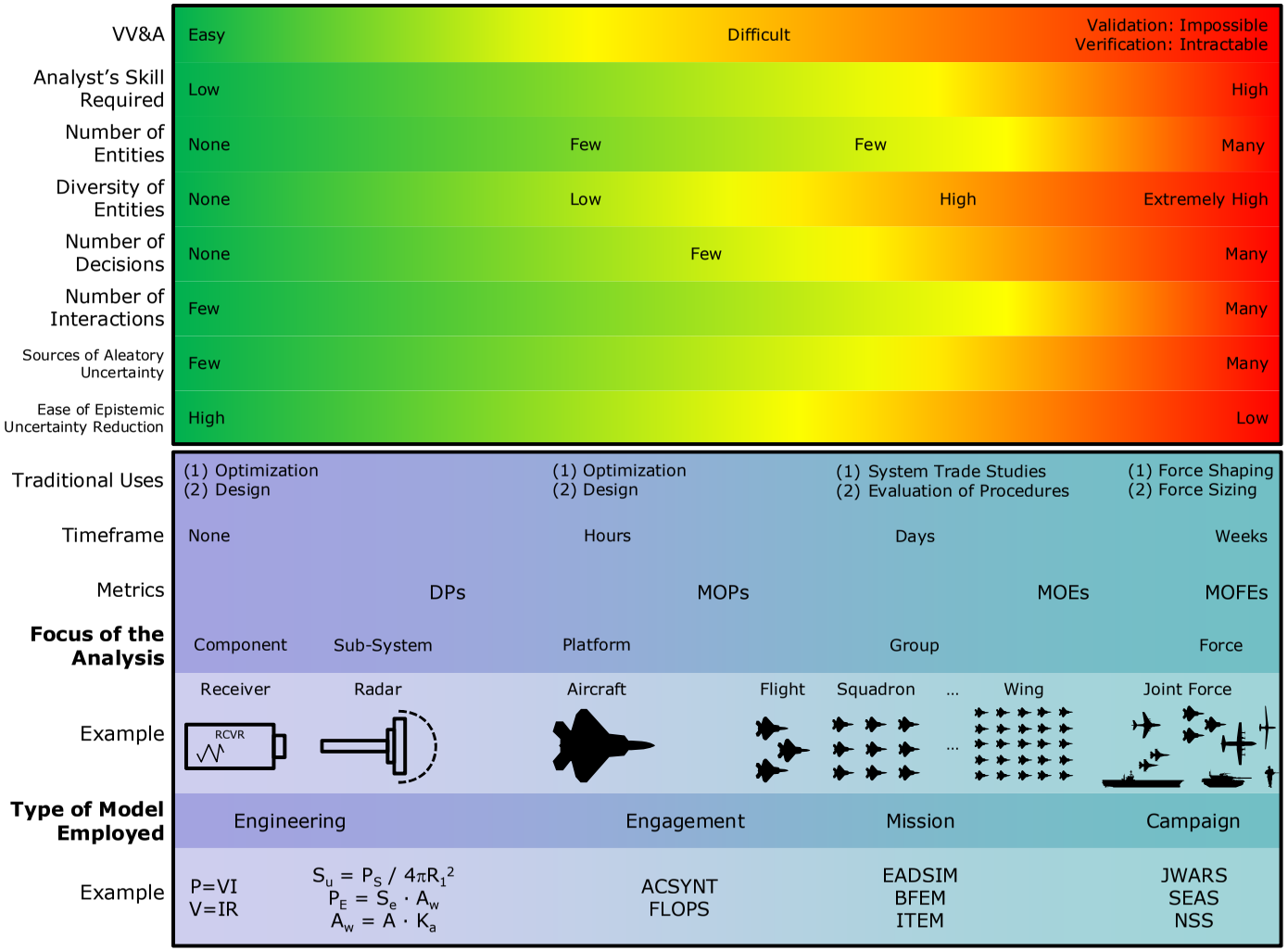

Figure 15: Continuum of Military Models and their Complexity.(Based on [305])

Ground Combat, (2) Air Combat, (3) Naval Operations, (4) Logistics/Personnel, (5) Air-Land, (6) Joint Operations, (7) Operations Other Than War, etc., etc. This taxonomy does not seem to offer a useful classification for the modeling theorists, but it has been extensively used by the DoD modeling practitioners interested in a particular application or tool since it offers them a quick way to scope the option space to that of their interest. For a comprehensive review of the models used by the military until the mid-1980s, refer to Battilega and Grange [48].

The Soviets had their own taxonomies of mathematical models of armed combat. In his 1974 edition of "Mathematics and Armed Combat," K.V. Tarakanov [318] expands on the differences between Soviet analytical and statistical models.([348], pp. 44) He states that analytical models are used for "simple" operations, and they tend to use a wide variety of mathematical techniques (e.g., arithmetic functions to differential equations) to relate system parameters to effectiveness criteria. On the other hand, statistical models, as employed by the Soviets, tend to leverage Monte Carlo techniques to capture the randomness in the operational combat processes. They use these when the processes modeled are "extremely complex, where a large number of parameters interact."[348] Tarakanov argues that the statistical models "possess substantive advantages over their analytical counterparts, and are devoid of profound assumptions and limitations."[348]

2.3.2.2 Decomposition of Modeling and Simulation

Despite the fact that no single, strict and widely accepted taxonomy of modeling techniques exists [141], significant efforts have been put forward to attempt to organize an ontology of modeling techniques. For example, professors Fishwick[6] and Miller[7] are currently collaborating on a project to create a M&S ontology by applying Semantic Web technologies [130]. Due to the scope of the project, they focused on Discrete Event Simulation for their Discrete-event Modeling Ontology (DeMO) ontology. They plan to use what they define to be upper level ontologies (i.e., for mathematical or statistical techniques) to create a hierarchy that will bridge abstract levels with more specific descriptions. A comprehensive ontology for

[6]University of Florida
[7]University of Georgia

M&S has not yet been defined, but work continues to be done [121, 129, 237, 295, 316]. Despite their efforts, their ontologies and taxonomies have not been extended to generalized modeling and simulation techniques as of yet, therefore for the purpose of this thesis a classification will be created based on the most comprehensive classification identified in the literature. The most comprehensive and detailed classification identified in the literature was described by Ron Ferguson [126] and is presented below:

Mental Models These are the cognitive representations (those that reside in the mind) and they form the basic decision-making tools. The difficulty of using these models is that due to the characteristics of the human thought process. Namely our limit in memory (only the effects of a few variables can be compared and tracked), our linearity of thought (effects that affect themselves and are interrelated with others are difficult to predict), and our use of subjective information (e.g., personal experiences, heuristics, etc.). Furthermore, mental models are difficult to formalize and communicate since they are of the most intangible form and often times we used them without explicitly thinking about it.

List Models These consist of enumeration of items, usually used for accounting (not only in the financial sense of the word, but in general) and projecting. The purpose is to assist the mental models with issues related to limits in memory, and enable the mind to track more variables and effects. The benefit of these type of models is that it is a familiar tool and generally the results are easy to analyze and understand. The difficulty lies in recording and analyzing the data, and the fact that the data may not exist in the form needed, i.e, the data to populate the lists may not exist and may have to be generated or inferred from other data. List models cannot help in performing this task.

Case Studies A method developed by the Harvard Business School in the 1920s. It consists of a detailed and broad account of a situation. By using expertise, a group iterates on the tradeoffs until consensus is achieved. It works as an iterative form of a group mental model and is meant to assist with the third difficulty faced by

mental models. The difficulty in applying this technique is its inability to account for large number of complex relations, the dependency on qualitative assessments, and the considerable time demanded to complete a study. The benefit is that it promotes understanding amongst the members of the study through discussion and the sharing of knowledge to build understanding.

Computer Models The advances in availability of computational power, as depicted in Figure 16, has fueled the advancement of the field of computer modeling. The significance of Figure 16 is that if the current trend in duplication of computational power every few years[8] continues, it is quite possible that by 2050 a machine with the same computational power as all of humanity combined can be purchased for the equivalent current value of one thousand U.S. dollars. The drastic increase we have seen in the past and most importantly, what is projected, have provided decision makers with the ability to create models that can become increasingly complex. The difficulty when creating these models lies in capturing the relations most critical to the problem at hand, and avoid creating a model that is so complex, that it does not contribute to the understanding of the problem at hand. Also, it is generally difficult to model relationships that cannot be easily quantified, or models that include a large number of intangibles. Nonetheless, computer models hold the key to bridging the three gaps of mental models. By being able to store and represent large volumes of data in a myriad of forms, it is possible to assess large volumes of information. The ability of the digital computer to perform an exceedingly enormous amount of operations per second, enable the study of nonlinear and competing effects. Finally, decision making support techniques can help decision makers extract their subjective knowledge and challenge it by testing it against the models.

An important concern when creating computer models often is what to include. The fact that computational power is increasing so drastically does not signify that the complexity of our models should increase at the same rate [221], merely that if more

[8]Note that this is not Moore's Law which stipulates the period for doubling the number of transistors in a computer chip.

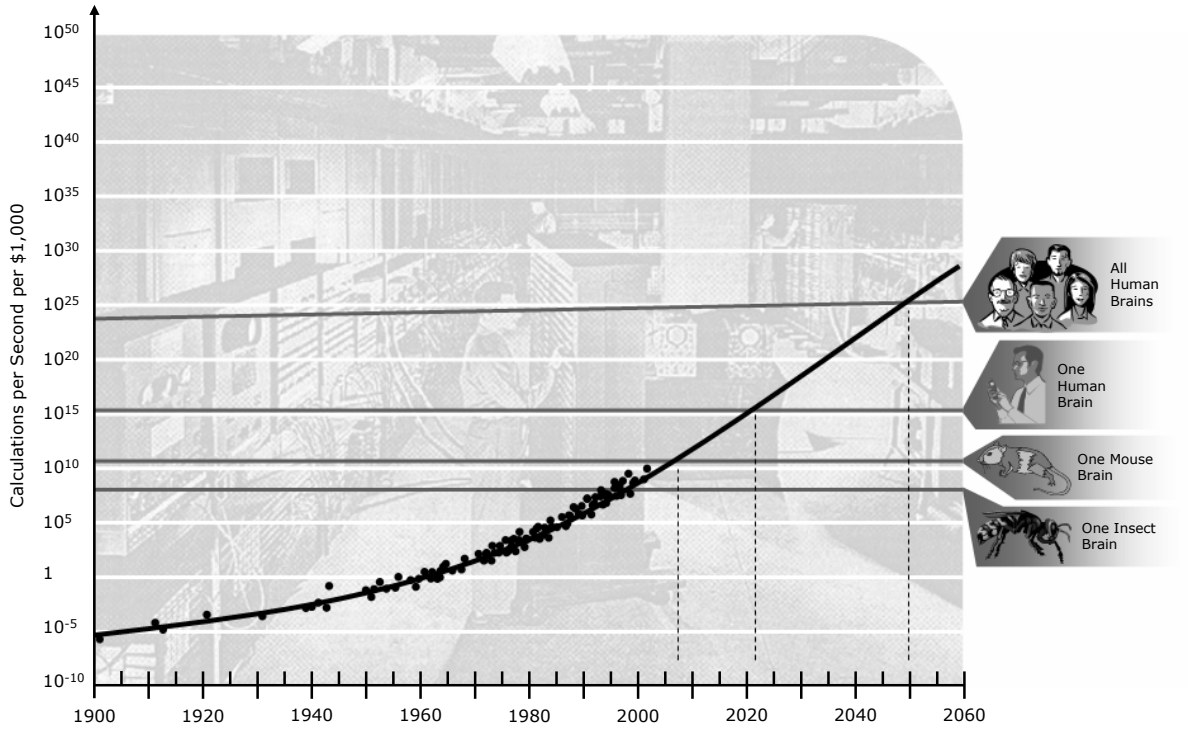

Figure 16: The advancement of the availability of computational power per unit cost [203].

computational power is necessary because the problem demands a more complex model, these requirements can be satisfied within a short period of time.

2.3.2.3 Computer Models

The first three types of models cannot capture large number of relations and interdependencies, a critical need for understanding complexity. For this reason, the focus will lie in the latter. A further analysis of computer models demonstrates that they can be classified in three categories.

Optimization Models These are models, often algebraic, developed for the purpose of determining an optimum setting of a system. Originally developed to solve problems in military logistics, they are closely related to operations research, e.g., traveling salesman, linear programming, etc. The problem with this type of analysis is that

it tends to be time independent, i.e., feedbacks and dynamics are generally not considered. This renders these types of models ineffective for the modeling of complex systems, since the dynamic behavior is critical to understanding emergence and the complex system itself.

Econometric Models This type of models are generally used in the social sciences and economics to analyze large volumes of data and draw conclusions with a certain degree of statistical certainty. The detriments are that it cannot distinguish between correlation and causality, they rely on large volumes of data, and in practice cannot predict behavior very well. Furthermore, for the purpose of analyzing complex systems, they do not suffice because they cannot explicitly model the relationships between the parts and require the existence of large volumes of data that in many cases do not exist when evaluating revolutionary, or even conceptual, architectures. Their strength is in the analysis of large volume of complicated data and the short-term prediction of the behavior of the system that created said data.

Simulation Models These are dynamic models developed to emulate the actual performance of a system. Used by a wide field of scientists, e.g., from the social scientists that use them to evaluate policies to the engineers that use them to evaluate the dynamical behaviors of systems they are tasked to design or control. There is a wide number of types of simulation models. The advancement of computational power, as previously, mentioned is enabling this field to grow exponentially fast. The importance of simulation to the point that Simulation-based Engineering (SBE) is being recongnized as a formal new form of engineering and has even been the focus of a recent National Science Foundation (NSF) blue panel study [257].

This last description implicitly defines the difference between simulation and generic modeling. Simulation in the context of modeling, implies the exercising of a model (a mathematical relationship between entities) over time. Therefore, a simulation to some degree, must have a dynamic (time-dependent) behavioral component. For the purpose of this research, behavior will be defined as the "manner in which a thing acts under specified

conditions or circumstances, or in relation to other things" [3]. The ability to study how something will act under a given condition, or when it is allowed to interact with other entities, is the key to building understanding and developing appropriate mental models that will enable proper decision making.

2.3.2.4 Simulation Models

> "Simulation refers to the application of computational models to the study and prediction of physical events or the behavior of engineered systems. The development of computer simulation has drawn from a deep pool of scientific, mathematical, computational, and engineering knowledge and methodologies. With the depth of its intellectual development and its wide range of applications, computer simulation has emerged as a powerful tool, one that promises to revolutionize the way engineering and science are conducted in the twenty-first century."
>
> - NSF Blue Ribbon Panel on Simulation-based Engineering Science [257]

Simulation Models may be used for a variety of things, as described by Kleijnen et al. [195], there are two extremes, modeling for insight and modeling for prediction. Modeling for insight is done when the mechanisms that drive a process or behavior are not well understood. These models tend to guide policy or select a particular course of action to mitigate an observed problem. Modeling to predict is done when a verified and validated simulation exists and can capture a previously unobserved situation. This is a particularly dangerous enterprize in modeling since only fairly trivial models can be truly validated.

The problem of what to model is critical and has been identified by many researchers and scientists [279, 334, 64, 56]. On the one hand it may seem useful to include as many aspects relevant to the system to be simulated as possible, but this hinders that ability to generate useful mental models from it. Or in the words of John Maynard Smith: "The better a simulation is for its own purposes, by the inclusion of all relevant details, the more difficult it is to generalize its conclusions..." ([56], p. 5) Model conceptualization is either a field or application specific craft, or an art.

Network Simulation Many models have been labeled "Network Models." For example, in the late 1960s Charles Bachman developed a database model named the network model [37], models of communication networks have been also named network models, e.g., Internet models, *ad hod* Network Model (NM), etc. For the purpose of this body of work, a NM will be defined as a model of pairwise relationships between entities of a certain collection. The model (or graph G) can be described as a set of entities (or vertices $V(G)$) and relationships (or edges $E(G)$). The relationships can be undirected or directed, and weighted, the entities may be divided in one or more groups, and the models can be layered to depict different types of relationships. In general the models are nondeterministic and must be solved using stochastic techniques, e.g., Monte Carlo Simulation, etc. Simple NM techniques have been extended to provide more comprehensive models for engineering applications [253, 60].

Dynamical Systems Simulation These are simulations of dynamical systems as described by ordinary differential equations, e.g., mechanical systems, electrical systems, hydraulic systems, etc. They are generally used to model a small number of components that have geometric and time interdependencies and are governed by ordinary differential equations [66]. This modeling technique is suitable for physical systems, but its applicability to complex systems is not as extensive as it is needed for the purposes of this research.

System Dynamics Simulation System Dynamics is a top-down modeling approach where the aggregated behavior is modeled directly. It was developed in the 1950s by Jay W. Forrester [132] with the intent of bridging techniques developed by the engineering controls community for applications to business strategy. In mathematical terms, a System Dynamics model is a system of differential equations. The key complex feature that System Dynamics models is feedback and its effect on the aggregate behavior over time [131]. System Dynamics has been used to model very large systems, even the entire aggregate behavior of the world to study growth policies and the complex impact they would have on the global population [127].

Discrete Event Simulation Discrete Event Simulation is a technique by which the behavior of the system is not explicitly tracked over time, but the states of the system are updated as events take place, i.e., the time-solver of a discrete event simulator does not integrate a set of differential equations as System Dynamics does, but randomly picks time sequences from distributions to simulate how the system changes state over time. It was developed in the 1960s by Geoffrey Gordon [152, 153] and has been favored by the logistics community to model supply chains [160, 77] and by industrial engineers to model assembly processes [330].

Markov Simulation A Markov Simulation is a simulation that abides by the Markov process, namely, that any state is only a function of the previous state, and is independent of any state prior to that. A Markov process is often referred to as a memoryless process, meaning it has no memory of its previous states. Also referred to as a Markov Chain Simulation, a Markov Simulation is a set of random variables having the property that given the present state, the future is conditionally independent of the past states. Originally developed by Metropolis et al. [235] and Hastings [163] using a Monte Carlo algorithm to simulate the Markov models. If the states can be enumerated, and the transitions between those states specified, a simulation can be conducted to determine how the system stochastically transitions between states, how often it is in a given state, etc. The difficulty in using these types of simulations for complex systems lies Markov chains demand the explicit enumeration of the states, which for a typical system of interest to an engineer, and a complex system in particular, is not feasible due to their combinatorial nature [331].

Petri Net Simulation Petri Nets were invented by Carl Adam Petri in 1962.[265] They were found to be suited to model distributed systems that exhibit synchronization and contention for physical resources.[38] Standard Petri Nets were more suitable for qualitatively modeling complex systems, but the addition of temporal effects, *Timed Petri Nets*, allowed for the quantification of the performance of real systems. Nevertheless, Petri Nets produced two serious drawbacks, (1) there were no data concepts

in the original formulations, this led to large models since the data manipulation had to be represented in the net, and (2) the formulations lacked the concept of hierarchy, this made the creation of models difficult. New developments since its original conception has extended the capabilities of Petri Nets, namely Colored Petri Nets (CPN) [331] and Stochastic Petri Nets (SPN) [38], have shown to be promising modeling techniques when studying reliability of complex systems [331, 38] and resource management [149]. SPN, as described by Volovoi [331], have the benefit over standard Markov chains is that SPN can be used to generate larger Markov models in an automated fashion, effectively bridging the inability of traditional Markov modeling to represent anything but problems of trivial scales. Nevertheless, CPNs have difficulty modeling inhomogeneous spaces [197] and adapting them to new processes oftentimes means that the models need to be discarded and a new one created from scratch.

Poisson Simulation Poisson Simulation is a stochastic extension of Continuous System Simulation, and in a sense it can be considered to be part of System Dynamics (in the sense that it solves a first order differential equation). At the same time it borrows similarities with a Markov Simulation and effectively enhances it by using aggregated state variables instead of state sets that must be explicitly enumerated in a Markov Simulation.

Poisson Simulation is useful when events happen one at a time, and the number of events during the interval is independent of both the number of past events and the times these events occurred. It was developed by Leif Gustafsson [157] in the late 1990s. Gustafsson claims that it models aggregates in a more efficient manner than Markov Simulation [158]. Poisson Simulations have the following characteristics and features [158]:

1. They are a system of differential equations can be described in terms of states and flows;
2. A change in a state value only occurs through inflows and outflows to that state;
3. The stochastics are located in the flow rates and not in states or parameters;

4. The stochastics are implemented so that the integration step-size Δt can be adjusted to the dynamic needs without distorting the model;
5. It can handle discrete states because the Poisson mechanism adds or removes integer numbers.

The multi-level modeling that Poisson Simulations can provide may be beneficial to the study of combat as a complex system, but its implementation is not clear. This is a fertile area for future work, but will be considered beyond the scope of this thesis.

Cellular Automata Simulation Cellular Automata Simulations consist of models composed of grids of cells, where each cell can have one of a number of finite states. The state of each cell is updated in discrete time according to a set of rules. The rules may depend on the previous state of the cell or its neighboring cells [228].

Originally developed by Von Neumann to explore the possibility of self-replication, cellular automata despite being simple to create have been used to understand problems as diverse as forestry [228], land combat [173], and chemistry [103]. Recently, Cellular Automata have been used as the primordial example of a system composed of very simple elements (finite state cells with simple deterministic rules) which when interrelated and exercised over time produce patterns far more complex than the rules used to create them [345, 346].

Nevertheless, Cellular Automata are not ideal for modeling the systems that are of interest for this research because the entities of interest posses more complex rules and have the ability to traverse through an environment, and are not rigidly related to the same static entities. Using automatons would demand extreme levels of abstraction.

Agent-based Simulation In a sense, Agent-based Simulations are a one-step evolution of Cellular Automata Simulations. Agent-based Simulations are time-stepped simulations of models comprised by agents who exist in an environment, interact with other agents and the environment, and make decisions based on those interactions. The definition of an agent varies depending on the source, but in general it can

be characterized as a myopic (can only perceive a portion of its surrounding state) self-controlled (can make decisions based on what it perceives) entity. Bankes [39] describes Agent-based models to be: "Agent-based Modeling (ABM) are examples of complex adaptive systems, which can be characterized as those systems for which no model less complex than the system itself can accurately predict in detail how the system will behave at future times."

Unlike System Dynamics and other top-down simulations, Agent-based Simulation (ABS) are bottom-up, meaning that the modeling concentrates on the parts (the agents) and their behavior, and then simulates a large number of stochastic runs to obtain the aggregate behavior, which can be emergent, i.e., displaying higher level patters that could not be inferred from the elements and the rules they operate by [173].

The great benefit of modeling using ABS is that the determining the basic logic and capabilities of the individuals is often simpler than determining the aggregates since these include nonlinear interactions between the agents and the agents and the environment. This allows modelers to specify simple entities and study their macroscopic behavior as it emerges from the simulation and do so without concerning themselves with aggregates.

The main shortcomings of ABS are that for large systems it is hard to determine what should be modeled and the large number of interactions render the Central Limit Theorem technically ineffective [73]. The consequence of the latter is that extensive number of cases must be executed to obtain statistical significance with any degree of confidence. Furthermore, the simulations can become exceedingly complex, demanding extensive runtimes.

The taxonomy described previously is presented as a tree diagram in Figure 17, with the selected techniques and subcategories colored in green. The reader is reminded that the techniques presented in the previous description are not all the computer modeling and simulations techniques developed. These are simply the most commonly employed by

analysts, and they capture the gamut of possibilities. Of all these M&S techniques, four were of particular interest because alone, can span the extremes of the modeling needs, these four can be considered to be "purebred" enough[9] to offer sufficient breath to study how these techniques can aid in the understanding of complex systems. The four techniques selected to be studied in more detail are Network Models, Discrete Event Simulation, System Dynamics, and Agent-Based Simulation.

2.3.2.5 Network Models

Network models are based on graphs, and make extensive use of theorems and algorithms developed by graph theoreticians. In essence a network model, or graph, is a set of nodes connected by edges. The nodes can represent entities (e.g., people, cities, military systems, etc.) and the edges represent a pairwise relationship between the entities (e.g., friendship, road connection, compatibility, etc.). The relations can be directional (directed graph) or bidirectional (undirected graph); the entities may be grouped into two or more exclusive categories that cannot share an edge, e.g., bipartite, tripartite graphs, etc.; and the edges may be weighted to provide a metric for the relations.

A recent focus in graph theory has been the development of random graphs to study the characteristics of large networks. Beginning with the work of Paul Erdös and Alfréd Rényi in 1959 ([252], p. 1) who first developed the concept of a random graph to study the properties of complex systems. As an example of the application of this model, researchers were able to understand how super clusters are formed, i.e., when water freezes it gradually forms islands of ice until it reaches a point that very quickly almost the entirety becomes ice, the mechanism for these sudden transitions were explained by the simple Erdös-Rényi random graph model. Since the development of these models, the literature has proliferated at a superb rate. Of particular interest is the work by Watts and Strogatz [337] and Albert and Barabási [16]. These researchers have developed random network models that demonstrated how certain characteristics of very large networks can affect their capabilities, behaviors, and mechanism of growth. New random models continue to be developed

[9] In the sense that they are not hybridized and approach the behavior of the model from distinct perspectives.

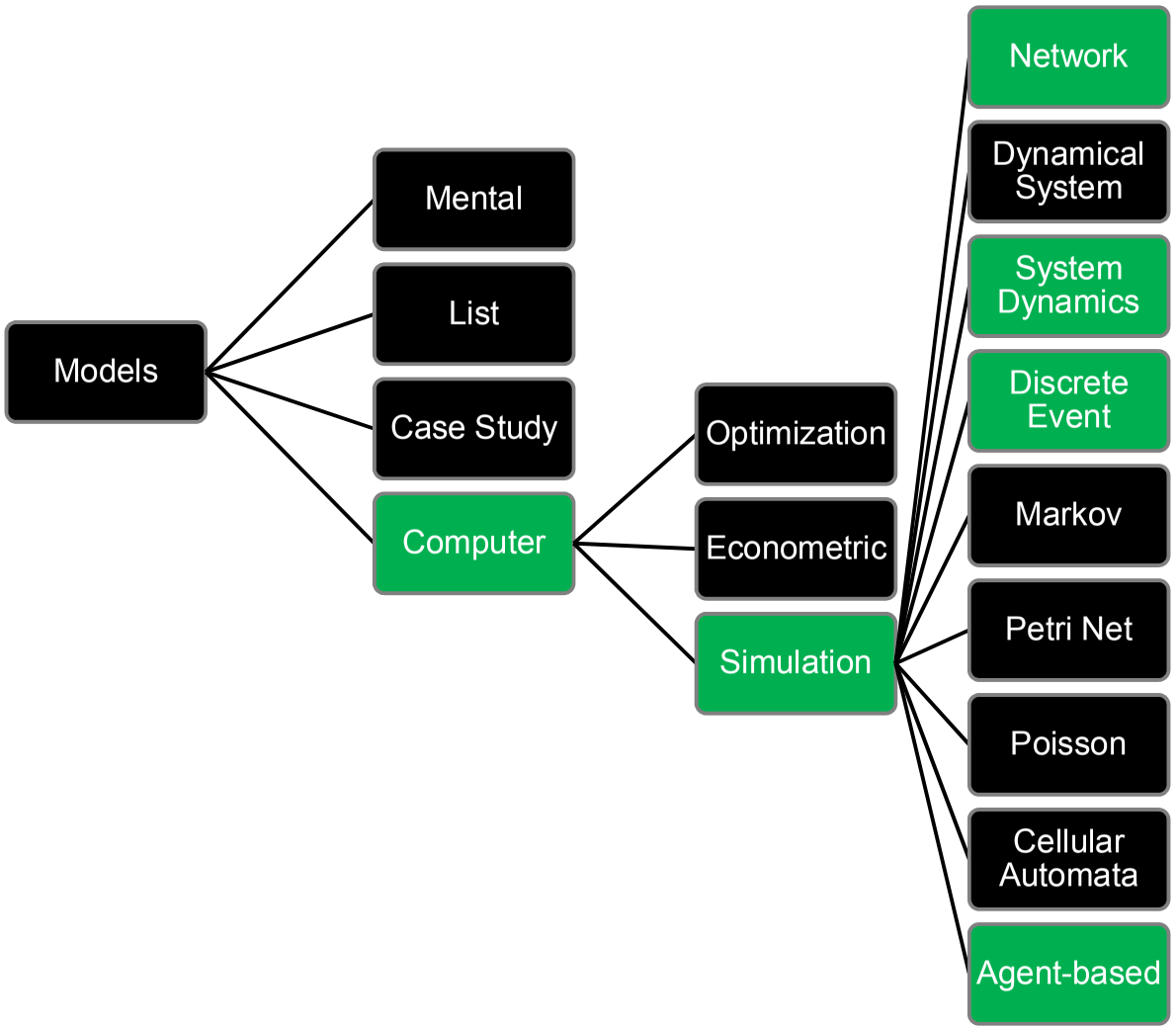

Figure 17: Modeling and Simulation Taxonomy.(Based on [126])

for specific applications, but these two models have prove to have characteristics that are universal to networks and have incubated the seeds for concepts like the small-world effect, degree distributions, clustering, network correlations, random graph models, models of network growth and preferential attachment, and dynamical processes taking place on networks. In the last decade, these have been the focus of extensive mathematical and applied research.[253] In recent years, researchers have "developed a variety of techniques and models to help us understand or predict the behavior of [complex] systems" ([253], p. 1). In his review of random graphs as models of networks, Newman [252] provides a thorough review of the literature. Weber and Porto [339] review the literature further and state that the research field of complex networks focuses on three things: (i) measuring network topology, (ii) investigating dynamics on networks, and (iii) studying the interplay between dynamical processes on networks and the network topology. This is of great encouragement for this body of work since understanding the relationship between topology (structure) and dynamics (behavior) is of critical importance to understanding the behavior of complex systems. In 2005 the National Academy of Sciences created a "Committee on Network Science for Future Army Applications" and published their findings and recommendations in a 124 page report titled "Network Science".[114] Below is a list of their three overarching conclusions.

1. Networks are pervasive in our modern world, and are vital to the workings of our economy and to the defense of the USA against both "conventional military threats and the threat of terrorism;"

2. "Fundamental knowledge about the prediction of the properties of complex networks is primitive;" and "Significant investment in the development of the core content of network science is required in order to create adequate knowledge to meet current demands for the characterization, analysis, design, and operation of complex networks;"

3. "Current funding policies and priorities are unlikely to provide adequate fundamental knowledge about large complex networks;"

More recently, military applications of network theory have been proposed and studied.[71,

180, 215, 284, 8] For example, Dekker and Colbert [97] propose that network analysis should be the main tool to analyze the robustness of military networks. They define optimal connectivity as the most robust topology the network can have given a minimum degree (d_{min}). Their methodology for creating robust networks is based on a metric they developed termed node-similarity, which measures how similar nodes are from a topological perspective. This is of importance for military networks, since trying to have homogeneous node-similarity makes the nodes equally important and decreases vulnerability. Comprehensive reviews of network generation algorithms have been developed by Weber and Porto [339], Albert and Barabási [16], Dorogovtsev and Mendes [109], and Newman [251]. Of particular interest, is the network generation method developed by Thadakamalla et al. [320]. Their method creates UltraLog network, a compromise between a scale-free and a random network to avoid the vulnerability of scale-free networks and maintain its robustness to random attacks. Their technique has been successfully applied to the creation of Distributed Multi-Agent Systems (DMAS).[62]

Several network models have been developed to study complex military systems. The following are just two of the most relevant examples identified in the literature.

Multiechelon Network Model for the CS4 The Multiechelon Network Model and Heuristic for the Combat Service Support Supply System (CS4) was originally developed to study the ability to lower the over-all inventory levels through more efficient transportation scheduling. The focus was centered around Class I supplies, in particular perishable rations. It used dynamic flows and a minimum-cost flow network algorithm to model the logistic system. It used a heuristic to determine the efficiency as well as feasible solutions to the multicommodity flows. It can also be used to study the reduction of pre-positioning inventory levels as well as the burden on transportation systems within a given theater. [353]

The Information Age Combat Model Developed by Cares [72], it is a graph theoretic approach to analyzing networked operations. The benefit is that it can analyze a large number of feasible networks and can be extended to study their time-dependent

behaviors. The Information Age Combat Model is generic in nature, in that it decomposes the entities that comprise a military force into four categories that can be aggregated to describe any existing military system.

Sensors These receive signals about observable phenomena and relay it to the deciders.

Deciders Receive information from the sensors and assess it to determine the present and future arrangements of other nodes.

Influencers Receive information from deciders and act on it to affect other nodes.

Targets These are nodes that are neither sensors, deciders or influencers, but still have military value.

The model serves as a test stand to understand how different network metrics impact the behavior of systems that more closely resemble military system-of-systems. It has been used to study the feasibility of novel logistic approaches, such as DAL.

Many measures have been proposed from the study of networks. Tables 3 and 4 contain some of the most interesting summaries found in the literature. Table 3 contains the rules of thumb proposed by Cares [72] for developing networked military architectures. The table contains Cares' recommended settings and the benefit of attaining that for 10 quantitative metrics that can be obtained using simple graph analyses. For a more detail explanation of the terms in the table the reader is referred to ([72], p. 101). Table 4 contains a list of the properties defined by Dueñas-Osorio [113] to determine the characteristic behavior of a resource distribution network. He uses these to characterize existing networks, and then uses the characteristics to create larger random networks which can be studied to infer how the self-similar larger networks will behave. Diaz et al. [104] proposed additional metrics, along with the algorithms to compute them, in their survey of graph layout problems. Mojtahedzadeh et al. [243] proposed their Pathway Participation Metric (PPM) (reproduced in Equation 1) as a way to assess the criticality of the links in a system dynamics model by studying their structure. They relate the criticality of each link, to the Perron-Frobenius

Eigenvector (PFE) and its associated eigenvalue, and the steady-state total participation metric to the the associated eigenvalue. These metrics and what they characterize are related to real systems characteristics. What is of interest to the network theorist, is the fact that despite the fact that graphs are highly abstracted representations, their characteristics are of value in understanding the real systems. Network models are one of the most extreme forms of models, in that they simplify and abstract to the maximum level possible, yet retain the essence of the system being studied, and can therefore help in understanding that system's behavior and characteristics.

$$PPM(i,j) = \frac{\frac{\partial f_k^j}{\partial x_i}\frac{\dot{x}_i}{\dot{x}_k}}{\sum_{i=1}^{n}\sum_{j=1}^{n(i)}\left|\frac{\partial f_k^j}{\partial x_i}\frac{\dot{x}_i}{\dot{x}_k}\right|} \tag{1}$$

2.3.2.6 Discrete Event Simulation

Discrete Event Simulation (DES) was formulated by Geoffrey Gordon in the 1960s [153, 152], for the purpose of understanding chronological sequence of events.[248] It is an approach based on the "concept of entities, resources and block charts describing entity flow and resource sharing" [59]. Several mechanisms have been proposed for conducting the simulation, the most prominent ones are: (1) Event-oriented, which describes the model states in terms of the consequences of events, programmed to describe "...how system state changes take place" ([192], p. 52), (2) Activity-oriented, which describes the actions of objects comprising the model and the conditions for these actions to take place, and (3) Process-oriented, which considers "... a set of events that are associated with a system behavior description" so as to combine "... the run-time efficiency of event scheduling with the modeling efficiency of activity scanning" ([193], p. 22). The algorithm to solve the simulation over time is global and it typically contains stochastic elements.[59] DES is tightly related to other modeling techniques, such as Markov Chains (Finite State Machine), Markov Process (Stochastic Process), and Petri Nets, which can be considered sub-categories of DES.

DES is typically spatially implicit, meaning that spatial relationships are not explicitly modeled because the models are more concerned with the study of the processes than the

Table 3: Thumb Rules for Analysis and Experimentation. [72]

Property	**Range**	**Effect**
Number of nodes, n	$n > 100$	Networked effects unlikely to occur with $n < 50$
Number of links, l	$l < 2n$	$l << 2n$, too brittle $l >> 2n$, too much overhead
Degree Distribution	Skewed	Adaptivity, Modularity
Largest Hub	< 100 links	Hub appears, recedes by reconnection of 5% of links
Characteristic Path Length	$\log(n)$	Short distances even for large networks (e.g., 10^4 nodes $\rightarrow$ Average Path Length = 4)
Clustering	Overall: $0.1 - 0.25$ Distribution: Skewed	Hierarchy, Organization
Betweenness	Distribution: Skewed	Highest: Most Important nodes, bottlenecks. Cascade Control [350]
Path Horizon	$\log(n)$	Self-Synchronization occurs around this value [328]
Coefficient of Networked Effects (CNE)	$0.1 - 0.25$	Measures the amount of cyclic behavior per node and compares the potential for networked effects in networks of different sizes
Neutrality Rating	$0.8 - 1.2$	Increased adaptation; decreased susceptibility
Susceptibility	Low (random removal) High (focused removal)	Hubs should be kept obscured until needed, damage abatement/repair schemes

Table 4: Fundamental Properties of Networks. [113]

Property	**Equation**	**Effect**
Mean Distance, L	$L = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d(i,j)$	Allows us to measure whether or not a network has the *small-world effect*
Vertex Degree, $d(v)$	$d(G) = \frac{1}{\lvert V \rvert} \sum_{v \in V} d(v)$	This plays a critical role in determining the fate of a network when subjected to random and targeted attacks [17]
Clustering Coefficient, γ_v	$\gamma_v = \frac{\lvert E(\Gamma_v) \rvert}{\frac{1}{2}d(v)(d(v)-1)}$	Measures how connected the network is in local scales
Redundancy Ratio, R_R	$R_{Rv} = \frac{1}{(\lvert S \rvert - 1)^2} \sum_{j \in V(\Gamma_v^2)} I(v,j)$	Captures the redundancy of the network at local levels
Efficiency, E	$E = \frac{\sum_{i \neq j} \frac{1}{d(i,j)}}{\sum_{i \neq j} \frac{1}{d_{ij}}}$	A global indicator of efficiency in network connectivity [206, 207]
Connectivity Loss, C_L	$C_L = 1 - \langle \frac{n_F^i}{n_F} \rangle_i$	Quantifies the decrease in number of generators with connecting paths to the distribution vertices
Service Flow Reduction, S_{FR}	$S_{FR} = 1 - \langle \frac{S_i}{D_i} \rangle_i$	Quantifies the amount of flow that does not meet the distribution vertex demands

spatially derived interactions between the elements. This has been a detriment for developers of military models since the spatial considerations are often critical to the behavior of the system. Buss [67] offers an alternative to modeling movement and detection using pure discrete-event simulations by implicitly determining the x, y position of entities, i.e., specifying their location as a function of time, e.g., for linear motion $x(t) = x_0 + v \cdot t$. Although they acknowledge that as the number of entities increases, the computational effort increases exponentially and the benefits of modeling using an event-driven approach disappear when compared to discrete time steps.

DES is generally the modeling approach desired when problems of queuing and bottlenecks are driving the performance of the system. Examples of areas that have used DES to optimize their products and processes are manufacturing, telecommunications, customer service, health, and airport transit. Relevant military examples are logistic simulators such as T.LoaDS [160], an expeditionary warfare model developed by the Naval Postgraduate School (NPS) [261], the Naval Simulation System (NSS) [124], . These models all share the common characteristic that in an effort to capture as many elements as possible, their results are often difficult to digest and the models become difficult to adapt.

2.3.2.7 System Dynamics

System Dynamics (SD) was formulated by Jay Wright Forrester in the 1950s [132]. He prescribed a graphical lexicon derived from control theory to model "soft science" problems, e.g., economics, ecological, sociological, etc. Golfarelli et al. ([150] p. 2) define System Dynamics as "an approach to modeling the behavior of nonlinear systems, in which cause-effect relationships between (aggregate and quantifiable) abstract events are captured as dependencies among numerical variables; in general, such dependencies could give rise to retroactive interaction cycles, i.e., feedback loops." Feedbacks are the critical feature of SD models and are the main artifact for creating the complex behaviors observed. Borshchev and Filippov ([59], p. 5) mathematically define a System Dynamics model to simply be "a system of differential equations." As differential equations, SD models are inherently continuous in time and in their pure form cannot distinguish between entities but can only

model high-level aggregates. Sterman [311] is recognized as one of the best sources on the subject of System Dynamics modeling.

An important feature of System Dynamics is that they operate at the aggregate level, meaning that the individual entities that compose a system, and their behavior, are not explicitly modeled. The model parameters are concerned with averages, or more appropriately aggregates, that describe the behavior of large groups of these entities. This makes it difficult to model systems that contain highly heterogeneous populations or models where the aggregates are not easily determined a priori. Borshchev and Filippov elegantly describe these two difficulties: (1)"As long as the model works only with aggregates, the items in that same stock are indistinguishable, they have no individuality," and (2) "the modeler needs to think in terms of global structure dependencies and has to provide accurate quantitative data for them" [59]. A third difficulty of SD is its validation and verification [271, 285]. The main difficulty is that obtaining macro-level data and values for aggregates is more often than not a labor intensive task. For this reason, researchers have been proposing novel approaches to validation of SD models through the study of their structure [271, 285] and through a method of simplification [118].

As with DES, researchers have been attempting to extend the basic capabilities of SD models by adding features and increasing their complexity. The most prominent example of this has been the work by Koopman et al. [200], who devised a series of models that sequentially increase the capabilities of traditional SD models. They formulated four different types of models which are listed below. It is important to mention that this model hierarchy makes use of both network models and DES models. Table 6 defines which assumptions are made by each model and what each of their characteristics are.

1. Differential Equations (DE) models are traditional SD models. The deterministic, infinite homogeneous population interacting homogeneously assumption is retained.

2. Stochastic Compartmental (SC) models relax the infinite population size assumption and include stochastic effects enabling the modeling of distributions of outcomes. They retain the assumption that prior histories do not affect the individual outcomes.

3. Individual Event History (IEH) models relax the compartmental assumption by allowing individuality between the elements and have the capability to track the individual histories of each entity. They are easier to construct than SC models because the combination of possible states do not need to be enumerated a priori.

4. Dynamic Network (DNW) models relax the homogeneous relationship between entities allowing for time-dependent relationships between the entities.

Regardless of the efforts of researchers, the fact remains that creating SD models is a labor intensive task that requires large amounts of expert knowledge to make the decisions as to what should be modeled and how ([155], p. 83). A customary practice is to use case studies to guide in the development of the models, but this makes the development of these models more of an art than a science. Studies indicate that the amount and quality of information given to a SD model user will severely affect their learning and success in achieving objectives ([155], p. 82). Since the purpose of this research is to help understand complex systems through modeling and simulation, the findings of Größler et al. are most interesting. Schaffernicht [287] argues that most studies that attempt to describe how SD can contribute to the understanding of complex behaviors focus on the models and not on the process of modeling. And the fact is that both mental models and explicit models are tightly coupled, since explicit models articulate part of the mental model and their development alters the mental model, but the explicit models (especially SD models) rely heavily in mental models and implicit knowledge. In SD there are a variety of models, e.g., Causal Loop Models (CLM), Influence Models (IM), and Stock-and-Flow Model (SFM), which are used to help the user create the appropriate mental model of the system at hand. The difficulty of measuring the effectiveness in enabling this transition is the querying of the mental model. Since they are by nature implicit, it is difficult to quantify them and translate them to explicit form. The tool commonly used to do this are Cognitive Maps (CM), but their capabilities are limited at best.

Table 5: Characteristics of four model forms. [200]

	DE	**SC**	**IEH**	**DNW**
Assumes infinite population size	yes	no	no	no
Assumes homogeneity within compartments	yes	yes	no	no
Assumes that contacts have no duration in time and that who is contacted is not influenced by history of prior contacts	yes	yes	yes	no
Contact rates can be defined by region	yes	yes	yes	yes
Allows one to evaluate study design and data analysis methods	no	partially	yes	yes
Models events deterministically	usually	rarely	rarely	rarely
Models events stochastically	rarely	usually	usually	usually
Ease of mathematical analysis	easiest	OK	hard	harder
Solved numerically	yes	no	no	no
Usually simulated for analysis	no	yes	yes	yes
Complexity of analytic results	simplest	OK	complex	most complex

Table 6: Modeling the elements. ([287], p. 79)

Element	CM	CLM	IM	SFM
Boundary	+	+	+	+
Time Horizon	-	-	-	+
Resources (stocks and flows)	-	-	+	+
Sectors	-	+	+	+
Policies	-	-	-	+
Feedback Loops	-	+	+	+
Delays	-	+	+	+
Nonlinearities	-	-	-	+

2.3.2.8 Agent-Based Simulation

"Agent-based models (ABM) are examples of complex adaptive systems, which can be characterized as those systems for which no model less complex than the system itself can accurately predict in detail how the system will behave at future times." Steven C. Bankes [39]

Agent-based simulation is a bottom-up approach, often used for exploratory purposes, where the researchers create simple entities with a set of rules to guide their behavior. These entities exist in an environment and can interact with one another either directly or indirectly. The definition of environment does not necessarily entail a spatial environment, it could be a virtual environment, e.g., agents representing simple routines interacting through the World Wide Web to simulate virus attacks, etc. The definition of "agent" is not universally agreed upon and researchers tend to define them as it best suits their purposes [54, 171, 213]. Some definitions include:

- "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" [282].
- "An *autonomous agent*[10] is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" ([139], p. 6).

[10]Emphasis in the original

- "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" [349].

- Essentially "a surrogate life form" ([54], p. 351).

Simpler definitions seem to survive scrutiny better, but provide less insight as to what an agent is. For the purpose of this work, an agent will be defined as a myopic (locally aware) entity, which interacts with other agents and its environment, and operates on explicit rules. Agent-based Modeling and Simulation provides a very broad modeling and simulation capability and can capture the largest number of characteristics of complexity. By working from the bottom up, Agent-based Modeling & Simulation (ABM&S) can provide a mechanism for studying the most complicated characteristic of complexity, emergence. The exploratory use of ABM&S consists of formulating a series of agents (including rules and properties) and an environment in which they interact and analyzing how their macro-level behavior develops. Applications of ABM&S have ranged from the simulation of escape dynamics [165], combat models [173, 208], civil transportation system [213], social systems [75, 224], and economics [121, 145].

These models have traditionally been used for providing new insight and increased understanding of the dynamics of emergent behavior, but increasingly, they are being used for prediction and simulation-based decision making [65]. This means that the models need to be verified, quantitatively calibrated, and validated. As with SD, the validation of ABM&S is difficult because despite the fact that data on the lower levels of the model is generally readily available, obtaining data on the macro-level behavior is labor intensive and in some cases, practically impossible. For this reason, macro-level results, which is the goal of ABM&S, are often not possible to validate, nor calibrate. Since strict validation by comparison to historical data is often not possible, new techniques to achieve this have been proposed. As with SD models, Dr. Qudrat-Ullah [271] proposes the use of structural validation as an initial step for validating ABM&S models.

Distributed Networked Operations Simulator Distributed Networked Operations Simulator

(DNOsim) was developed by Jeff Cares and his collaborators at Alidade Inc. [72] to study revolutionary fighting concepts from a highly abstracted perspective. It is currently being reinvigorated by a group of researchers from industry, government and academia. The model extends the Information Age Combat Model discussed in the network modeling section. Its current implementation is in NetLogo, but its highly abstract nature allows it to be easily ported to other frameworks.

Enhanced Isaac Neural Simulator Tool Enhanced Isaac Neural Simulator Tool (EINSTein) is an extension of Irreducible Semi-Autonomous Adaptive Combat (ISAAC) developed by Andrew Ilachinski [173] to study the nonlinearities of combat and provide a more insightful model than the traditional Lanchester equations of attrition. His model is based on agents that abide by simple rules that depending on a handful of parameters (how many friendlies are close by, how many enemies, where is the objective, etc) make the agents maneuver, engage, or flee. Despite its simplicity, EINSTein produces highly complex behavior, clearly demonstrating the ability of ABM&S to generate emergent behaviors.

2.3.2.9 Benefits of Agent-based Modeling

The big advantage that ABM has over other modeling paradigms is that it enables the analyst to study how the elements impact the macro behavior by only knowing how the elements interact with one another. Agent-based models are a small computational petri dish, where analysts can infuse the agents of their choice and watch them interact and learn from those interactions. Philosophically they hold one foot on the deductive realm and one on the inductive. They are deductive in nature because the agents operate from a set of assumptions (methods and properties, rules and states) and their behavior is derived from these assumptions. Yet, their purpose is to aid in inductive reasoning. They offer analysts the ability to describe the better known phenomena (the lower level phenomena) and observe how the aggregation processes produce the less known phenomena. None of the other methods can perform this aggregation. It is true that the rules and characteristics programmed by the analysts must be true for the observations to be valid, but this is

nonetheless, the only one of the four techniques described that can accomplish this modeling feat. In some instances, it is possible to create highly adaptable ABMs that allow analysts to test various alternatives under a variety of scenarios. These are ideal for building mental models, and heuristics about the behavior of systems, something that can be invaluable when studying a system that has never been observed, conceived or understood before. Furthermore, ABMs can allow analysts to develop intelligent agents that display learning traits, which can aid in the modeling of complex systems and in understanding how complexity arises. In essense, ABMs' greatest benefit, in particular when compared to the other three alternatives described, is their ability to capture characteristics of systems directly easing the conceptualization of the models of complex systems.

2.3.2.10 Detriments of Agent-based Modeling

The principal detriments of ABM is its computational expense and the "art" required in knowing what should be modeled and how. Furthermore, constructive simulations using the ABM paradigm can require extensive efforts to develop, verify and validate.[355] Some authors have argued that it is never possible to truly validate an agent-based model,[72] although some have done some remarkable efforts in attempting to validate such models.[82, 121] It is important to mention that these two bodies of work relied on copious amounts of data that may not be available when creating a constructive simulation of an architecture that does not exist.

$$N = \left(\frac{ts}{\alpha \bar{x}} \right)^2 \tag{2}$$

A final detriment which is associated with the computational requirements of ABM&S is that when assessing risk, the analyst needs to perform a stochastic simulation of the dynamic activity.[355] This can be problematic if the values need to be resolved with sufficient resolution, but the range over which it needs to be resolved is proportionately small when compared to the absolute value. By Student's t-Test, which assumes a normal distribution so its results must be weighed accordingly, one can estimate how many cases (samples) are required to calculate the mean within a range from the true mean with a given certainty.

Table 7: Student T-Test for outputs of a Simple Stochastic Agent-based Model. [72]

	Blue Killed	**Red Killed**	**Blue AAM**	**Blue ASM**	**Red AAM**
Std Dev of the Sample	0.757	4.231	6.088	10.126	0.940
Average of the Sample	0.743	15.604	31.683	9.257	0.762
Cases for 10% accuracy	407.43	28.82	14.47	469.04	595.58
Range for 95% certainty with 30 samples	36.85%	9.80%	6.95%	39.54%	44.56%

For the model used in the demonstration presented in Table 7, a 101 repetitions were executed and the data used to provide the inputs to Equation 2, where N is the number of cases required, t is the t-parameter which is a function of the number of samples used (for 101 samples and 95% certainty, t equals 1.98), s is the standard deviation of the sample, α is the accuracy required as a fraction (e.g., 10% accuracy would be represented as 0.1), and $\bar{x}$ is the average of the sample.

The results in Table 7 show that for some measures of the population, e.g., Blue Killed, the amount of repetitions required to obtain 10% error in the accuracy of the prediction of the true mean (which is not extremely high, but it is a small margin for small values) with 95% certainty (a decent confidence interval, but once again, not as high as it may be desired), may be unattainable. This is a particularly pertinent problem when attempting to create surrogate models from these stochastic dynamic simulations, where the error in estimating the mean with a handful of cases (e.g., 30), may be over 40%. Kleijnen et al. [194] discuss the importance between replicates and "scenarios," which they define as a different vector of inputs, e.g., a design point. They argue that if the purpose is to find a robust system, then more replicates are desired and at least some are essential. On the

other hand, if the goal is to understand or compare systems, Kleijnen et al. argue that if a constant variance can be assumed, or Common Random Numbers (CRN) can be used to reduce the variance, then emphasis should be put on the variety of design points or scenarios evaluated. A problem with ABM is that implementation of CRN tends to "move the simulation entities away from the definition of agent."[81] An interesting approach to use ABM and obtain statistical significance without doing multiple sampling, is that of Parunak and Brueckner [262] who employ a "polyagent" approach. The polyagent approach consists of avatars (the original agents) and ghosts that explore their alternative behaviors. The avatars hatch ghosts as they interact with others and the environment. A swarm of ghosts is maintained by each avatar, the cardinality (number of) of which can remain constant or within some ranges by killing older ghosts and hatching new ones. The ghosts do not interact directly with one another or with other avatars but indirectly through a digital pheromone field. The environment keeps track of the pheromone field as a vector of scalars. The biggest hurdle with this approach lies in the assumption that the alternate behaviors of the agents can be described by the pheromones, which is something that the authors acknowledged is "open to questions."

The motivation for Parunak and Brueckner is a very clear concern when using ABM&S, in that it can easily become prohibitive to ensure that sufficient samples have been taken to sample the possible alternative interactions thoroughly. If $n+1$ entities are modeled, at each step each entity can interact with n other entities. As Parunak and Brueckner explain, at time t the history of interactions for each entity $h(t) \in n^t$. By its definition, the behavior of an agent depends on its interactions with other agents, i.e., $h(t)$. Since ABM&S explicitly models the trajectory of each individual entity, it cannot study the aggregates directly as Equation-based Modeling (EBM) does, and therefore, individual runs must be repeated to obtain insight into the averaged or aggregated behavior of the system. If the model is executed for τ time steps, each agent will experience one of n^τ possible histories, where for a population of $n+1$, only $n+1$ of these histories will be sampled (assuming homogeneity of entities and initial conditions). Furthermore, simulations tend to be executed for more time steps than agents, i.e., $\tau >> n$, meaning that even for homogeneous simulations with

low sensitivity to initial conditions, only a small portion of possible histories tends to be explored.

2.3.3 Evolution of Combat Modeling and Simulation

Until recently, mathematical combat models relied on some form of the Lanchester Equations. Frederick Lanchester was a British mathematician who attempted to apply mathematical analysis to warfare.[204] His basic law is reproduced in their simplest form in Equation 3, the initial quantities ($B(t = 0)$ and $(R(t = 0))$–representing the quantity of forces the blue (B) and red (R) sides have at time $t = 0$–must be greater than zero for the equations to produce useful results. The equations relate the rate of attrition of a force to the strength ($B(t)$ and $R(t)$) and individual effectiveness (k_B and k_R) of the opposing force. The equations presented below are most often referred to as Lanchester's Linear Equations, which were developed by studying one-on-one combat. In these equations, the rate of change in a force dB/dt and dR/dt for the blue and red sides respectively, is related to the quantity that both sides have and an attrition multiplier the attacker has, $-k_B$ and $-k_R$ for the blue and red sides respectively.

$$\begin{aligned} \frac{dR}{dt} &= -k_B B(t) R(t) \\ \frac{dB}{dt} &= -k_R R(t) B(t) \end{aligned} \tag{3}$$

In these equations, as one side looses units, its ability to inflict damage on the enemy is reduced. At the same time, as a side looses units, its rate of loss is reduced. Lanchester realized that his main assumption for his original model was that one unit could only engage *one* other unit, e.g., sword combat, spear combat, etc; and that fire was undirected. This agreed with the fact that as a side's force was reduced, its rate of loss would be decreased, rate of loss was proportional to both the quantity of the enemy and the quantity of the friendlies. This was a suitable assumption in ancient combat, but modern combat was characterized by directed and undirected fires. To amend this, he modified his model to what has been often referred to as the square model, from which the square law was derived. In this adaptation, the linear law equations are modified to become the ones presented in

Equation 4. In this case, the rate of loss is only proportional to the size of the enemy force, indicating, that if fewer friendlies are still alive, the enemy will find them with equal ease and eliminate them.

$$\begin{aligned} \frac{dR}{dt} &= -k_B B(t) \\ \frac{dB}{dt} &= -k_R R(t) \end{aligned} \tag{4}$$

It was later realized that these equations are just special cases of a more general equation presented in Equation 5. The general case reduces to the (1) linear law if $\phi_1 = \phi_2$, (2) the square law if $\phi_1 = 1$ and $\phi_2 = 0$, and what is termed the (3) logarithmic law when $\phi_1 = 0$ and $\phi_2 = 1$.[344] Hartley [162]–using combat data from the Korean War–argues that a mixed model with $\phi_1 = 0.75$ and $\phi_2 = 0.40$ produces the best results. These models have in general been used to answer the following questions:

$$\begin{aligned} \frac{dX_1}{dt} &= -\theta_1 X_1(t)^{\phi_1} X_2(t)^{\phi_2} \\ \frac{dX_2}{dt} &= -\theta_2 X_1(t)^{\phi_2} X_2(t)^{\phi_1} \end{aligned} \tag{5}$$

- Who will win?
- What force ratio is required to win?
- How many survivors will the winner have in a battle of annihilation?
- How long will it take to annihilate the enemy?
- How do the number of combat forces change over time?
- What is the sensitivity to force (e.g., $\frac{B(t=0)}{R(t=0)}$) and kill (e.g., $\frac{k_B}{k_R}$) ratios?

Extensions to the Lanchester attrition model are still being studied and developed to this day (e.g., [335, 161, 83, 221, 33, 185]), although their applicability to modern warfare (in particular its application to information age warfare) seems to be less accepted.[286, 72] The models are continuous in nature and therefore are only applicable to the engagement of

large forces, for asymmetric combat, the model as originally conceived by Lanchester is not deemed applicable because the underlying assumption that rates of attrition are proportional to force sizes is violated. Among the most notable modifications are (1) heterogeneous models, (2) discrete models, and (3) stochastic models.

1. Heterogeneous models separately consider the force of each system type, and the attrition is a function of the sum of the force of the opposing systems that can eliminate that given system.

2. Discrete models integrated in discrete time-steps

3. Stochastic models where the effectiveness parameters are not a single value but a distribution.

A generalized Markovian model based on probability theory–which simplifies to the Lanchester equations under special conditions–was proposed by Helmbold [166] in 1966. Helmbold's model is actually heavily influenced by Robertson's [278], who studied survival probabilities between several targets and shooters. Helmbold's model studied intervisibility effects and differences in weapons by modeling volleys of shots. Helmbold defines a set of shots to be a volley "if and only if each of the separate shots is directed against a target chosen independently and at random from among the available targets, and attrition information is withheld from (or ignored by) the shooters until termination of the set of shots."

The homogeneous modeling of combat by the Lanchester-based techniques has been recognized as not capturing the effects that the diverse types of units have on the outcome of a battle. Colonel Dupuy and his associates [116, 117] developed Quantified Judgment Method of Analysis (QJMA) and later Tactical Numerical Deterministic Model (TNDM) to address the effects that different types of weapon systems have on the overall force's ability to win a conflict. His method is characterized by a large number of parameters that attempt to capture weapon capabilities, effects due to weather, terrain, training, decision making, dispersion, etc. The models were calibrated with data on battles between the 19th

Century and the 1970s.[115] The method has been extended to include modern combat parameters and has been compared to combat statistics from the First Gulf War [117] and continues to be extended to this day by military analysts.[185] Despite some promising results, the method is characterized by a large number of parameters that at times seem arbitrarily set, and it is not clear how much of the model is over-fitted to the data. A similar Soviet development to QJMA was Correlation of Forces and Means (COFM).[151, 348] The main difference was that COFM was focused in supporting operational commanders in deciding what force ratios were required before attempting to subdue a defending force. This regulated and well known operational construct hindered Soviet maneuver and was often simplified by the NATO members as the 3:1 force ratio, whereby a Soviet commander would not attack until he had attained a 3:1 force ratio over his adversary.[72] Nonetheless, the current implementations of TNDM and COFM do not explicitly handle spatial properties of combat, interactions between the different combat systems, decision making, and the network of the force.[117, 185] These are critical to today's combat and are predicted to be at the core of future force's capabilities.[91, 41, 72, 78, 79]

Military models are composed of three elements [270]: (1) weapon effects, (2) human behavior, and (3) decision-making. The focus of models in the past has been on weapon effects, mainly through attrition. The other two aspects are much more difficult to capture, but are just as important in determining the outcome of a conflict. Traditionally, the lack of ability to model these aspects has been bridged by the use of wargaming. Nonetheless, wargaming is an expensive preposition and limits the number of different scenarios and perturbations that can be executed.

For this reason, new emphasis has been given to employing different modeling and simulation paradigms for analyzing combat operations. Ilachinski [173, 171, 172] was one of the first to postulate that the modeling methods of the past did not apply to the new era. His two models ISAAC and EINSTein were employed to provide insight into the nonlinearities and emergent behaviors that may be observed in the battlefield. Ilachinski's work is focused on capturing the effects of human behavior by using simple artifacts that reproduce behaviors observed in real combat situations. The concept of capturing more

advanced decision making require more advanced techniques. The field of AI has been identified as a viable area of research to bridge this gap, but the cost and inability to truly reproduce the ingenuity and adaptability of human decision makers, have in general given the proposed solutions a cold reception. For examples of applications attempting to give modeled military decision makers artificial intelligence, the reader is referred to [54, 138, 143, 282, 283, 294, 322, 349].

2.3.4 Campaign-Level Modeling Tools and Environments

When modeling military campaigns, the goal is to predict the outcome of a campaign (losses, cost, military objectives reached, etc.) as a function of forces (numbers, individual capabilities and their ability to coordinate), and the tactics they employ. The models in the literature tend to be focused on a force-on-force simulation, but that is not the only type of application nor their only purpose. Perla [264] argues that models of warfare can be identified as far back as 5,000 years ago with Wei-Hai, the Chinese wargame developed in 3,000 B.C., believed to be similar to Go. Chaturanga, an Indian game that would serve as the inspiration for the game of chess, was developed in 500 A.D. These were classic example of games used to train soldiers and generals. The Romans used sand tables and miniature replicas around 30 A.D.[303] The first modern wargame, Koenigspiel, was developed by the German Christopher Weikhmann in 1664. It consisted of a checkered board, similar to chess, with 30 pieces representing different military ranks. From this sprung War Chess and the more famous Kriegsspiels, whom the Prussian Chief of the General Staff, Baron von Muffling, is quoted as saying: "This is not a game; this is training for war. I must recommend it to the whole Army!"

The following frameworks tend to attempt to do this using a constructive (agent-based) approach, and they tend to be stochastic, in that events occur according to probabilities, which makes it necessary to run a number of simulations to obtain statistical significance of the outcome of a campaign.

2.3.4.1 Integrated Theater Engagement Model (ITEM)

Integrated Theater Engagement Model is a scalable (from small unit to Major Theater of Operations (MTO)) theater-level simulation. It was originally developed by the Navy's Defense Nuclear Agency as a naval, nuclear simulation model. In its evolution, it has been extended to include naval surface and subsurface combatants, amphibious operations, air combat, and ground combat. It is currently being further developed and maintained by the Defense Threat Reduction Agency (DTRA).

2.3.4.2 THUNDER

THUNDER was developed by the United States Air Force (USAF) to serve two purposes, (1) conduct analysis of the contribution of systems, capabilities, forces and employment of concepts within the context of theater-level operations [324], and (2) support near-real time wargaming. It can model thousands of entities simultaneously (e.g., aircraft, air-to-air and air-to-surface weapons, air defense sites, satellites, ballistic missiles, airbases, a variety of ground targets, as well as intra and inter-theater logistics).[225] It is composed of two main elements, a simplified (1) Ground War, and a more detailed (2) Air War. The Ground War portion is based on the US Army Attrition Calibration (ATCAL) Model. It is a time-stepped, deterministic model, normally calibrated at the division level. Combat is modeled as one-dimensional pistons for the purpose of analyzing force-on-force engagements between Warsaw Pact forces and NATO. The Air War is a stochastic, event-based simulation that relies on a large number of probabilities for almost all aspects of the air-to-air, air-to-ground, and ground-to-air combat modeling.[225] The probabilities are functions of a large number of parameters, e.g., the probability that an aircraft survives an attack is a function of pre-strike intelligence, the properties of the opposing force, night/day and weather, etc. The model aggregates agent behavior horizontally also for the air-to-air modeling. Models in THUNDER require careful creation of the probability databases to ensure that useful results are obtained from the analysis.

2.3.4.3 FLexible Analysis, Modeling, and Exercise System (FLAMES)

FLexible Analysis, Modeling, and Exercise System (FLAMES) is a framework developed by the Ternion Corporation for creating constructive simulations [54, 7, 319]. Using FLAMES a user can specify the characteristics of the objects in the simulation, the methods they employ, and their interfaces to *construct* the macro-system. The framework is itself composed of a series of products, (1) FLAMES Operational Requirements Graphical Editor (FORGE), (2) FLAMES Interactive Runtime Executable (FIRE), (3) FLAMES Scenario Highlighter (FLASH), and (4) FLAMES Analysis and Reduction Environment (FLARE). The engine of the simulation is FIRE, the other products are there to support the creation of the scenarios (FORGE), visualize the outcome (FLASH), and post-process the data (FLARE).

There are three types of models in FLAMES, (1) physical models (e.g., vehicles, sensors, jammers, communication devices, etc.), (2) cognitive processes (i.e., those that simulate human reasoning and decision making), and (3) models of natural and manmade environments.[319] FLAMES has been used for a variety of applications–e.g., network centric system analysis, conceptual weapon systems design, directed energy weapons performance analysis, and military engagement analysis–but due to the nature of the applications, many of the models and databases remain classified.[54]

2.3.4.4 Joint Integrated Contingency Model (JICM)

Also known as the Joint Integrated Campaign Model,[94, 137] this tool was developed by the RAND Corporation to analyze blue-on-red campaign-level engagements. It was created to assess (1) the future of warfare, (2) emerging trends in national objectives, (3) weapon technologies, (4) doctrines, and (5) force capabilities in both a Major Theater War (MTW) and a Smaller-Scale Contingency (SSC).[136, 260] It aggregates individual units, e.g., ground units are aggregated into brigades or divisions, air units are aggregated into packaged sorties. The entities and their interactions are represented with low level of fidelity. "The Joint Integrated Contingency Model (JICM) is a very large simulation system that encompasses the strategic and operational levels of land, air, and naval warfare with a global set of models and databases."[342] The modules included in the latter versions include

the following sub-models: (1) strategic mobility, (2) air combat, (3) land combat, (4) naval power projection ashore, (5) amphibious/air-mobile/airborne insertions, (6) ballistic/cruise missiles and theater ballistic missile defense.[136]

2.3.4.5 Joint Warfare System (JWARS)

The purpose of Joint Warfare System (JWARS) was to develop a state-of-the-art, constructive simulation that would (1) provide a multi-sided, balanced representation of joint theater warfare, and (2) use Command, Control, and Communications (C3) and Intelligence, Surveillance, and Reconnaissance (ISR) as the foundation for how entities perceive and interact with one another.[234] While "most large scale combat models have used only the relative capabilities of weapons and the number of each type of weapon involved in battle to determine the outcome of an engagement," JWARS "also considers the quality, unique capabilities, vulnerabilities, and perceptions of the leaders and individuals in units contributing to the outcome of every battle." [32]

JWARS "is an event-stepped simulation that describes the behavior and interaction of military forces across the joint spectrum." [314] It attempts to capture the effects of (1) a 3-D battlespace, (2) terrain and weather, (3) logistical constraints on the force performance, (4) key information flows, and (5) perception-based C2. [314]

Figure 8 contains some of the tools employed by the DoD to conduct campaign-level analysis. JWARS was created to bridge these and enable true joint campaign analysis under a common framework.

At the core of JWARS is the Battle Space Entity, which can perform 6 functions, (1) C2, (2) sense, (3) manage resources, (4) move, (5) communicate, and (6) own other Battle Space Entity (BSE). It has three levels of analysis for evaluating the outcome of a campaign, (1) strategic mobility, (2) theater logistics, and (3) joint warfighting.

The computational requirements of JWARS are substantial, and based on the literature, depending on the processing and data storage requirements, the cost to develop an infrastructure to support JWARS could range between $35,000 and $500,000. [296] The cost required to maintain this infrastructure could be considerably higher.

Model Name	Vintage	Users	Sample Applications	Land	Air	Maritime	Strategic Mobility
TACWAR	1960s	COCOMs, Joint Staff, OSD PA&E	Base Force, BUR, MRS BURU, Nimble Dancer (ND), Desert Storm, MRS-05, Operational Availability	●	◒	○	○
THUNDER	1980s	USAF	USAF Analyses, ND, JAST	◒	●	○	○
JICM	1990s	Joint Staff, OSD PA&E, US Army	Operational Availability, Transformational Force Assessment (TFA), TAA	●	●	◒	●
CEM	1960s	US Army	Army Analyses, TAA	●	◒	○	○
GCAM (ITEM)	1990s	US Navy	Navy Analyses, ND, Investment Balance	◒	◒	●	○
MIDAS	1980s	COCOMs, Joint Staff, OSD PA&E	Base Force, BUR, MRS BURU, ND, Desert Storm, MRS-05, Operational Availability	○	○	○	●

Table 8: DoD campaign simulation tools.

2.3.4.6 APL Integrated Multi-Warfare Simulation (AIMS)

APL Integrated Multi-warfare Simulation (AIMS) is an architecture to integrate other tactical/mission simulations using the High-Level Architecture (HLA) standards. The motivation for AIMS was to satisfy the perceived need for multi-warfare analysis[201]. The drivers for this requirement are listed as (1) transition to "capability-based" acquisition, (2) the creation of multi-mission structures (e.g., Sea Shield), and (3) the need to assess the performance/effectiveness of multi-mission platforms (e.g., DDG-1000).[29]

AIMS is not necessarily a simulation framework, but an integrator of other simulations, for the purpose of achieving joint multi-mission analysis. Its level of focus is slightly below the campaign level, but it encompasses more than just a single mission assessment. The creators have therefore modified the military analysis pyramid to include "multi-warfare" between campaign and mission.

2.3.4.7 System Effectiveness Analysis Simulation (SEAS)

System Effectiveness Analysis Simulation (SEAS) is a multi-mission (or "campaign slice"), two-sided, force-on-force, Monte Carlo combat model with explicit sensor geometry and information network topology constrained by circuit time delays. It was developed specifically for long-range planning of force structure acquisition strategy and, in particular, space-based C4ISR systems (command, control, communications, computer, intelligence, surveillance, and reconnaissance). Each agent in SEAS runs a parallel execution thread and interact with each other on discrete time steps. The process that each agent executes is based on Boyd's Observe, Orient, Decide and Act (OODA) loop.[102]

"SEAS is a PC-hosted, many-on-many, stochastic, theater-wide, multi-mission-level model. It is typically used for military utility analyses of present and future space systems to explore combat outcome sensitivities to Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) (Command, Control, Communication, Computers, Intelligence, Surveillance, and Reconnaissance) operational concepts and force structures. SEAS is a member of the Air Force Standard Analysis Toolkit listed on the [Air Force] Portal for use by the [Air Force] analytic community."[324]

2.3.4.8 Naval Simulation System (NSS)

NSS is the Navy's primary next generation simulation for operations, analysis, wargaming, and experimentation–NSS is the primary model for supporting network centric Fleet Battle Exercises [102]–its original purpose was to support operational commanders in developing and analyzing a Course of Action (COA) at the mission, group, and force levels.[26] NSS is an object-oriented Monte-Carlo, multi-warfare/C4ISR, discrete event M&S tool that simulates all Naval warfare areas (both at sea and shore-based) with moderate to high resolution.[124] It was originally developed by SPAWAR PD-15 and Metron, Inc. for Chief of Naval Operations (CNO) N6M. NSS explicitly represents the chain of command and models operational plans, including tactics, doctrine and situation-dependent actions. NSS "provides a comprehensive force-on-force modeling and simulation capability."[312]

Current work is focusing on extending it to include logistics enhancements. NSS explicitly models surveillance, communications, tactical picture processing, engagement, sea-based logistics, and Command and Control including plans and tactics, so that dynamic Expeditionary Logistics can be set up in NSS without extensive new software developments. Given the NSS representation of sea-based logistics, logistics C2, wargaming, and training applications can be conducted in the context of realistic scenarios, avoiding the risk of mismatches that are inevitable when different tools are used for C2, wargaming, and training.

2.3.5 Modeling and Architecting

Maier and Rechtin ([226], p. 18) state that at high levels of complexity, abstract techniques must be called into play, because purely analytical techniques can be overwhelmed. They argue that designers faced with these type of problems should resort to architecting techniques, experience-based heuristics, abstraction, and what they denominate "integrated modeling." The goal is to concentrate on the essentials, "consolidate and simplify the objectives... stay within the guidelines."[226] The system must be abstracted (modeled) at a high a level as possible, and this abstraction progressively reduced throughout the design process. Maier and Rechtin go on to argue that modeling is "the centerpiece of systems architecting," and that it is a "multipurpose, progressive activity, evolving and becoming

less abstract and more concrete as the system is built and used."

These are strong statements from two of the most prolific thinkers in the field of systems architecting. The common denominator between modeling and architecting is the fact that both rely mainly on abstraction. This is a process that requires high levels of cognition, where the critical effects and their driving causes must be identified and captured. Maier and Rechtin [226, 227] argue that this need to abstract, employ heuristics, and infer through induction is one of the key reasons why architecting is more of an art than a science. In contrast, they argue that engineering is more of a science, concerned with analysis, where decisions are based more on deduction from principles and theories.

The difficulty is then that if architecting relies heavily on induction and heuristics, when faced with a problem that requires revolutionary systems, the heuristics and the induction may not yield the expected results since they are by definition extrapolating into uncertain territory. Maier and Rechtin argue that the architecting of revolutionary systems has often been driven by a particular technology, that the project manager takes a leading role and overshadows the project architect, and that often times, the saving grace of the system introduced is not its intended application but a happenstance that was unprecedented in the original phases of development. These are not encouraging statements for those that pursue radical solutions to unprecedented problems, but Maier and Rechtin make a good argument that their observations have often been the rule and seldom the exception. Studying the development of high-risk aircraft validates their point, e.g., the F-117 focused on stealth technology, the A-12 (later the SR-71) on high-speed flight, the F-111 on swing-wing technology.

The crux of the problem lies in the extrapolation of abstractions and heuristics. The solution to this problem is often an excruciatingly costly Research and Development (R&D) program to obtain insight into the unchartered fields. Good examples of this are commonplace in the aerospace community, e.g., the experimental aircraft series starting with the Bell X-1 to understand supersonic flight, all the way to the more recent Boeing X-52 to understand hypersonic (Mach 7+) flight. The hope is that this cost can be curbed with the use of models, but all models must be validated and used within the ranges for which they

were validated.

The use of architectures within the DoD–and other U.S. government agencies–has been mostly as a record keeping and knowledge-base repository. Department of Defense Architecture Framework (DoDAF) in particular does not mandate any simulation–or any other analysis processes for that matter–to assess the goodness of the architectures represented.[239] In recent years, researchers have recognized and identified that the information contained in these architecture products can be used for analysis-based support of decision making. Of particular interest to this thesis are the extensions and adaptations made to the DoDAF that demonstrate how architecture products can be used to populate models [49, 102, 144, 239, 240, 354, 355], and show how those models can in turn analyze in a quantitative manner the worth of the different architecture alternatives. The most notable efforts identified in the literature are presented below.

Executable Architectures Levis et al. [11, 212, 211, 333] were the first to propose the use of models to assess the goodness of an architecture. Their approach proposes creating CPNs out of standard translations of architecture products, in this case the Integrated Definition 0 (IDEF0) specified in Federal Information Processing Standard (FIPS) Publication 183.[212] Their application is particularly well suited to information processing architectures, which is in accordance with their goal of modeling and evaluating C4ISR architectures, not general architectures.

Executable Model for Communication Architectures Baumgarten and Silverman [49] proposed using Extend™to implement discrete event simulations of DoDAF architectures by implementing executable products. In particular, they employ the SV-1 and SV-2 to evaluate network performance. Their focus is on the Information Technology (IT) aspects of the architectures, in particular by employing MCS to identify bottlenecks and design constraints early in the design process. Their future plans include interfacing their Extend™model with OPNET®and Telelogic's System Architect to increase the fidelity of their results and the automation of model generation.

Executable Architecture Methodology for Analysis (EAMA) This is an effort to ease the creation of models from standard DoDAF products. Their goal is to (1) dynamically analyze a system or capability, (2) measure the performance of a process, and (3) measure effort and resource utilization over time,[355] with a focus on IT architectures. Developed by the Pawlowski, Barr, and Ring from the Mitre Corporation,[355] the implementation chosen employs Petri Nets to analyze the operational architecture and an "OPNET-like tool" to analyze the measures of performance of the architecture's communication network.[144, 276, 355]

Discrete Event System Specification (DEVS) extension to DoDAF Mittal et al. [239, 240, 354] have proposed extensions to the standard DoDAF products to assess the performance at the Operational View level. The proposed DoDAF products would allow for automated DEVS model generation, which in turn can provide measures of performance and conformance. Conformance, as defined by Mittal et al., is the error margin for "departures from required behavior." For a given communications architecture, with a given set of objectives in terms of communications delays, the automated products can assess what is the likelihood that the architecture will meet the goals by studying a large set of possible scenarios. Zeigler applied such a methodology to the test of Link-16. The two products that Mittal et al. proposed to add to DoDAF are an OV-8 (activity components document) and an OV-9 (activity interface specifications). The OV-8 lists the activities as components with port interfaces. The OV-9 contains the information for the interface between activities and entities.

ABM Models Zinn [355] postulated that DoDAF products can be used to populate ABMs. For his particular application he used SEAS as the ABM framework. DeStefano [102] collaborated with Zinn but his goal was focused on evaluating the architecture, whereas Zinn's goal was to demonstrate how DoDAF products can be used to generate ABMs. Their thesis focused on ABM as the modeling paradigm because

their questions of interest centered on the evaluation of the goodness of C4ISR architectures, which as they identified in the literature–e.g., Gonzales et al. [151]–can only be captured by a modeling framework that can capture the individual entities' behavior and decisions.

2.3.6 Limits of Modeling and Simulation

There are some limits to M&S that often make it more of a craft (or an art), than a science. As pointed out by Rainey ([273], pp. 12) and others [13, 15, 233, 249], you can't capture every reality with models and simulation, nor should you try to since then the model becomes useless.[72, 221] This need to abstract inherently demands the making of assumptions, and these assumptions are the ones that may produce incorrect solutions or conclusions. More often than not, these assumptions are not stated explicitly by the modelers, increasing the risk of the incorrect application of a model.

Another common complication is the collection of proper data to help create the model and consequently validate it. Validation consists in ensuring that the model accurately represents reality, the difficulty of this cannot be overstated and it is rarely done properly or thoroughly. Since the decisions based on the models can only be as good as the models, and the model's validity can only be as good as its validation, improper validation has critical consequences, and it is often the Achilles heel of a modeling effort. Some researchers argue that it is not possible to truly validate some kinds of models, e.g., a large-scale constructive simulation, because no data exist to do so, and if there were, the complex effects taking place would indicate that the one or few existing data points are small samples of a larger set of possible evolutions of the model. The constructive simulation may produce a distribution of possible outcomes, and reality may only be one possible value within that distribution. Reality may in fact be any value within the distribution of possible trajectories that the system may take, and not simply the mean or some other statistically significant measure. If this is the case, the reader may wonder what is the purpose of creating models of such systems, and that is a good question indeed.

Creating models of complex system can demand large amounts of resources and time.

The execution of these models can be equally expensive, in particular for models which were deemed to be necessarily more similar to the system in question than abstract. It is important to remember that the primordial need for modeling and simulation has been to reduce cost in prototyping, therefore, it is important to curb the resources required by the model. Nonetheless, the more capable techniques, those that can capture more characteristics of complex systems, are the ones that require the more effort.

The repeated misuse of models has been one of the central reasons why many decision makers do not trust analytical models.[170] It is often the case that a model that was created for a particular application, or purpose, is extended and used for others to which it is not applicable or relevant. This is related to the previous point that ill defined assumptions in models may lead to risk in their application. This is only compounded by the lack of awareness of this and the blind trust afforded to some legacy or well-established models.

2.4 Synthesis of Findings

Campaign-level modeling for the analysis of (1) doctrinal solutions and (2) materiel solutions to capability gaps, as well as the (3) evaluation of novel strategies and (4) command and control paradigms has become more widespread. The DoD has been investing sizeable resources for the development of M&S capabilities, and trends indicate that it will continue to do so. Novel design methods for systems whose goodness is not a measure of their performance, but the measure of effectiveness of the overall force of which they are part of, also rely on a campaign-level type of model to conduct the analysis. With an insufficiently capable model, these methods suffer the traditional Garbage-In Garbage-Out (GIGO) problem, and their results become meaningless. Since these methods require the successive execution of a large number of different cases, analysts cannot afford to employ models that are excessively demanding to execute and verify.

A large body of work exists on the subject of modeling frameworks/techniques. Nonetheless, no ontology has been developed that is flexible enough to encompass all the techniques, and at the same time is specific enough that it can segregate the different techniques to a large number of groups, each containing only a few similar techniques. Of all the ontologies

surveyed, the unpublished one proposed by Ferguson was deemed to be the most applicable to the different paradigms considered. The decomposition of the techniques into the different branches led to the identification of a specific set, namely: (1) Network Models (based on the concepts of Graph Theory), (2) System Dynamics, (3) Discrete Event Simulation, and (4) Agent-based Modeling.

Figure 18 provides the benchmarking of the four simulation techniques surveyed in more detail with respect to the the characteristics of complexity and their ability to represent each, as well as the cost to develop and validate models in the given frameworks. The clear result is that no one technique is ideal. On one extreme there is the option of creating models that are easy to establish but cannot capture the characteristics of complexity as thoroughly as is the case with Network Models. On the opposite extreme are Agent-based Models which have the capability to capture the most complicated characteristics of complexity but demand large amounts of effort to create and may not be possible to validate. DES and SD lie between these two extremes in terms of capability and cost.

This inability to capture all the characteristics of complexity with a single model was to be expected. There are four possible solutions to the effective modeling of complex systems:

1. Develop a new modeling technique
2. Create a hybrid of existing techniques
3. Extending an existing technique
4. Integrate different techniques to create a modeling process

The first alternative offers the greatest flexibility but disregards the great strides that have been accomplished by extremely talented thinkers and researchers. An additional practical detriment of this option is that convincing the M&S community to adopt yet another technique will be difficult, if not unfeasible due to the large investments in training and tools. To counter this argument, a technique that is simple, leverages existing data, and can be learned without copious amounts of effort may be a feasible candidate. When observing how different modeling techniques evolved it is clear that no modeling technique

Method / Evaluation Criteria	Network Models	Discrete Event Simulations	System Dynamics Models	Agent-based Models
Nonlinearity				
Interactions				
Intelligent Agents				
Represent Hierarchies				
Emergent Behavior				
Adaptation				
Dynamic Behavior				
Ease of Creation				
Ease of Validation				

DEFINITION OF CRITERIA

Nonlinearity	Ability to model disproportionate causes-to-effects.
Interactions	Ability to model the effects of interdependencies between entities.
Intelligent Agents	Ability to model sentience of the entities.
Hierarchies	Ability to represent the organizational hierarchy of the entities.
Emergent Behavior	Ability to provide insight into the macroscopic behaviors that cannot be elucidated from the analysis of the individual entities in isolation.
Adaptation	Ability to model the capability to change of the individual entities, closely related to the ability to model sentience.
Dynamic Behavior	Ability to model time-dependent effects and changes of state.
Ease of Creation	How much time and effort must be devoted to developing a model.
Ease of Validation	How much time and effort must be expended in validating the models.

Figure 18: Benchmarking of different simulation techniques.

was developed for the purpose it eventually served out of nothing. For example, network models arose from the application of computers to mathematical tools (graphs) which were well established tools, SD was the product of the application of control theory to soft-science intangibles, Cellular Automata (CA) was the product of a mathematical tool, similar to graphs, that when coupled with the use of computers, enabled the study of more extensive problems, DES followed the same principles using probability, queuing theory and improved computational power, Petri Nets were developed by C.A. Petri for his thesis [265] as a tool to study simple communication phenomena, but were extended considerably decades later by researchers that observed that they could be applied to broader problems when integrated with probabilistics (SPN) and computer simulations (CPN). In summary, (1) it is difficult to permeate a new technique through a well established community unless that technique is extremely simple, useful, and has a low learning curve, (2) seldom has a successful modeling technique/method been initially developed for that purpose, they are generally evolved from simple applications that grow in applicability as more researchers adopt them and find value in them.

The second option has been extensively pursued by a variety of researchers with varying degrees of success. Hybrid simulations between SD and ABM&S have been argued for [288, 290] and achieved [59, 289]. The detriment of this particular hybrid is that it is not truly addressing the shortages of ABM&S, namely, it is not easing its development or Verification and Validation (V&V). CA and SD have been used concurrently to model a city's land use [164]. In this example an SD model was used to simulate demand and a CA model was used to simulate supply. It is not a hierarchical modeling process, but a co-simulation process. This approach of hybridization would involve selecting the most appropriate modeling technique for each portion of the complex system and prescribing some rules for creating those models and co-solving them over time. The difficulty of co-solving (employing separate time-domain solvers) simulations is the inability of ensuring stability and convolutes V&V further still. The best example of a true hybrid of techniques is proposed by Koopman et al. [200] who develops a series of hybrid models that range between strick system dynamics, to discrete event simulation, to networked discrete event

simulation.

The third alternative involves acknowledging that the existing techniques do not satisfy all possible requirements and that in order to do so, they must be extended or enhanced. A good example of this approach is the work by Buss and Sanchez [67] who recognize that DES cannot address spatially explicit events and therefore they formulate a method for translating continuous spatially explicit information into discrete events. Their work requires a considerable amount of effort, namely explicitly calculating the spatial trajectories of every entity and listing the times when they will intersect, this process needs to be repeated throughout the simulation every time an event occurs, which can considerably slow down the simulation. A second example of this approach is the work by Bounova and de Weck [60] who propose an enhanced network model that will allow to study more features of an architecture by incorporating more domain specific knowledge while still being able to study the network properties of the system. They state that their "method does not claim to solve the modeling problems of all systems imaginable, and it does require hard additional work for application adaptation." This relation between more modeling capabilities and increased requirements in modeling effort is observed in a third example [177] of an extension of modeling technique as well.

The fourth and final possibility involves developing a process that exploits the strengths of each modeling technique available and uses them to bridge the detriments of the others. More explicitly, network models with their ability to model breath more easily than ABM, can be used for the initial stages when the conceptualization of the large-scale system model is most difficult. Their inability to capture dynamic effects can be bridged by the use of SD or DES models of the portions that contribute the complex dynamical behavior to the system. And the spatially implicit nature of the dynamical modeling tools can be bridged by using the more costly ABM&S to determine how the spatial dependent metrics aggregate and can be fed to the dynamic simulation.

The four options are not mutually exclusive in that a process of established techniques could also include a new technique, or a hybrid. The hybrid of existing techniques could be augmented by introducing a new technique or paradigm. The fourth option seems to

be the most sensible given the large body of existing M&S techniques, nonetheless, a more thorough analysis of the most commonly used paradigms will indicate that constructive simulations using agent-based formulations are gaining more traction in the community. These are the more capable as previously described, at the same time, these are the more demanding, and the ones that seem to be conducted in an ad hoc manner.

When attempting to understand complex systems, "it is necessary for a person to "build a picture" of the structure of the problem incrementally" ([222], p. 39). It is not advisable for the modeler to immerse himself/herself in the most complex and demanding implementation possible, Maani and Maharaj's recommendation is not only sensible, its crucial to developing a model that will not only be useful but ready in time to be useful. As Captain Doerry from the US Navy often times reminded audiences, "the right answer late is not right at all."

The use of models should balance the minimization of effort with the maximization of fidelity or insight. As Downey explains, there are two types of models.[110] There are "reductionist models," whose primary virtue is realism, and there are "holistic models," whose primary virtue is simplicity. Despite the objective or the underlying philosophy of the model, the minimization of effort expended in creating and exercising the model always considered crucial and the counter balancing goal to creating the best model possible. For reductionist modeling, both the creation and the exercising of the model can be demanding (as in the case of a very large constructive simulation), for holistic modeling, since the goal is to create a simple model, the creation of the model may be difficult, requiring a large number of iterations to capture the behaviors of interest.

Two central objectives are drawn from the series of observations presented in the previous section:

1. To obtain insight into the behavior of large-scale system architectures without resorting to full-scale constructive simulations.

2. When constructive simulations must be created, to (1) determine quantitatively which entities should be modeled; and (2) whose behaviors should be modeled in more detail.

Rather than formulate research questions, which would be how to achieve these objectives, the objectives will be addressed by hypotheses. This will be the focus of the next chapter.

CHAPTER III

HYPOTHESES AND TESTING

> "I think that only daring speculation can lead us further and not accumulation of facts."
>
> - Albert Einstein

The purpose of this section is to explain how the research objectives posed previously will be answered. A series of hypotheses will be proposed from studying the literature on tools and techniques from other fields that may prove suitable to address the research objectives. The hypotheses will attempt to abide by Karl Popper's [269] interpretation of the hypothetico-deductive method and his definition of scientific hypothesis, namely, that it may be not only *practically* testable, but also falsifiable. The addition of the term practical is meant to differentiate between hypotheses that are practically falsifiable and theoretically falsifiable. For example, the statement: "it will snow in Atlanta on the 1st of January, 50,000 A.D." regardless of its usefulness, may be falsifiable, but it is not practically falsifiable since it will require us to wait till that date to ensure its validity. The concept of falsification will be used to ensure that the hypotheses proposed are scientific and abide by the scientific method, i.e., if a hypothesis cannot be falsified, no amount of experimentation can prove it wrong and no knowledge is gained from it.[268] An example of such hypotheses are generally those that include the word *can*, since any statement of the form: *this can be done* cannot be falsified because short of testing all possible scenarios where that can be done, and demonstrating that none can be done (something practically impossible) those hypotheses are not falsifiable. As a final note, the author would like to highlight that the tests described have not been conducted prior to the formulation of the hypotheses, augmenting the probability of showing the truth behind them. The subsequent sections will concentrate on presenting experiments that will support or disprove these hypotheses, since it is not possible to categorically prove through an inductive process the truth of a

hypothesis.[268]

The hypotheses proposed in this thesis will not include a method-level hypothesis. The main hypothesis of dissertations concerned with developing a method or process to aid in the design of a system tend to be of the form: "the way of doing something (e.g., designing) will be improved by the use of the process proposed in this thesis" (e.g., [142, 232]). This type of hypotheses are difficult to disprove or support, since in order to do so, an experiment involving experts must be be conducted to test the results of two groups, one using the state-of-the-art or benchmark method and the other using the proposed method. An experiment of this magnitude would be excessively costly due to the amount of experts' time required. A simplified demonstration of the method can be tested with volunteer subjects, but the validity of the results would be questionable, and a discouraging result, does not necessarily invalidate the hypothesis since it may be argued that the conditions did not reflect those expected in an actual implementation. For these reasons, this work will focus the scientific endeavor in formulating statements that will bring knowledge and understanding to the processes involved. The question of whether the proposed method(s) is/are better or worse than the existing ones will be left to the discretion of the reader.

3.1 Description of Hypotheses

The goal of this thesis is to intelligently expend modeling effort to obtain the maximum amount of insight into the behavior (produce understanding) of the system architecture of interest. As it has been explained in previous sections, most of this effort is focused in developing large-scale campaign-level constructive simulations. Therefore, the overarching goal can be divided into two distinct objectives:

1. To obtain insight into the behavior of large-scale system architectures without resorting to full-scale constructive simulations.

2. When constructive simulations must be created, to (1) determine quantitatively which entities should be modeled; and (2) whose behaviors should be modeled in more detail.

The first objective's goal, to obtain *insight into the behavior* of large-scale system architectures needs to be defined in detail. Constructive simulations are used to map lower level metrics (e.g., at the system level, measures of performance, etc.) to campaign-level metrics (e.g., attrition ratios, satisfaction of force-level capabilities, etc.). Furthermore, the simulations are used to compare two or more architectural options (including the operational aspects, the systems that compose them, and the performance of those systems) in their ability to achieve a goal (a capability) for a given cost. The first objective will be then reformulated for the application of this thesis to be "to quantitatively compare two or more large-scale system architecture's capabilities without resorting to full-scale constructive simulations."

The second objective will not be reformulated but for clarity will be expanded on to ensure that the ideas are not left up for interpretation. It may be possible that when analyzing large-scale system architectures, constructive simulation will be required since the solution proposed by hypothesis A may not be sufficient for all cases, or that additional analysis of a subset of the architecture still requires the use of a constructive simulation to fully understand its behavior. For this reason, it is important to determine how this constructive simulation should be created. This topic does not seem to be explicitly discussed in the military modeling literature, although the few examples that discuss how modeling is done seem to rely on a considerable amount of discretion from the analysts and modelers.[54, 134]

These two objectives will be addressed by two *testable* hypotheses. The first one focuses on quantifying the ability of a complex system architecture to fulfill a capability. The second focuses on guiding the modeler as to how the modeling effort should be expended.

Hypothesis A The spectral characteristics of the functional graph of a complex system architecture are correlated to that architecture's ability to fulfill a capability.

Hypothesis B An entity's centrality in the functional graph is correlated with its modeling criticality, where criticality is related to that entity's contribution to the behavior of the overall architecture, and therefore deviations in its behavior will have a larger impact on the architecture.

These two high-level hypotheses will be referred to from hereon as *Hypothesis A* and *Hypothesis B*.

3.1.1 Hypothesis A: Studying an Architecture's Capability with Spectral Graph Theory

The goal of this approach is to compare in a quantitative manner the ability of large-scale architecture alternatives to complete a capability.

In order for the technique to be useful, it must be possible to qualitatively estimate the Engagement Generation Matrix (EGM). In order for this to be possible, a single scenario must produce a unique EGM. If this was not the case, it would mean that there are unmodeled effects when employing the DiMA technique and the characteristic behavior of the scenario cannot be captured by it.

Hypothesis A.1 DiMA quantifiably compares the ability of large-scale complex system architecture alternatives to satisfy a capability requirement.

Hypothesis A.2 For a given scenario, the EGM has low variability, indicating that the EGM can capture the intrinsic characteristics of the scenario.

3.1.2 Hypothesis B: Focusing Modeling Effort on Functionally Central Systems

Models of large-scale architectures are generally created by modeling the behavior of the entities that compose that architecture, where behavior is mainly composed of performance and decision making. Decision making is in turn based on logic and is codified as a set of rules the agents must follow. Modeling the performance of the entities (i.e., mapping their measures of performance to their dimensional parameters and the environment) is not trivial, but is a small fraction of the effort when compared to modeling their decision making. Modeling the decision making of the entities involves modeling the rules by which they operate (i.e., capture their problem solving ability in algorithmic form). Traditionally, ABM&S simplifies this step to the minimum complexity rules required, or employs analogies to produce representative behaviors.[69, 173] But when creating constructive simulations for predictive purposes, i.e., creating models for the sake of being able to predict behavior, this

approach fails, since in essence, the agents are representing real entities that operate under an extensive set of rules. Human decision making is further clouded by issues of irrationality, errors, etc., which can be difficult to model or capture correctly.[34, 54, 305, 306]

This hypothesis will relate the functional structure of the architecture, i.e., how the different systems in the architecture are related to each other through their functional relations, to how much effort should be expended in modeling the rules of each entity. The first sub-hypothesis postulates a relationship between the *functional centrality* of a system within an architecture and its contribution to the overall's architecture's behavior. The second sub-hypothesis stipulates that focusing the modeling effort on the most central nodes will produce models that have higher fidelity (by having a lower probability of errors in the rules of the most critical entities).

Hypothesis B.1 The functional centrality of the elements that compose a complex system architecture is correlated to their individual impact on the architecture's behavior.

Hypothesis B.2 Focusing modeling effort on the functionally central elements of non-chaotic systems produces the highest fidelity models.

CHAPTER IV

MODELING THE CAPABILITIES OF COMPLEX SYSTEM ARCHITECTURES

> "The key observation that makes an abstract theory of networks possible is that the topology of those interactions–the 'architecture' of the network–can have a systematic influence on network behavior that is more or less independent of the precise details of the agents and their interactions"
>
> - Ian Stewart ([313], p. 601)

This chapter will expand on Hypothesis A, formulate a technique to test its validity, and test that technique's ability to achieve the goal set forth by the hypothesis, namely, to be able to *compare different architecture's ability to fulfill a capability* by studying the *spectrum of their functional graphs.*

4.1 Digraph Modeling for Architectures

This section describes the DiMA technique, a process for obtaining insight into the behavior of large-scale system architectures through the study of their functional structure. The technique was developed by the author to test the hypothesis that the spectral characteristics of the functional graph of an architecture correlates with the results from an ABM&S of those same architectures and is therefore suitable for comparing different architectures in terms of their individual capabilities. The purpose of this method is to quantitatively support the *comparison* of different system architectures' ability to satisfy a capability in a rapid, affordable and defendable manner. DiMA attempts to leverage existing architectural constructs and requirements, effectively postulating a method for integrating existing DoDAF products. This can be considered a possible avenue for making DoDAF executable, but the reader is reminded that that is not the main intent of the method, this method was developed to test Hypothesis A.

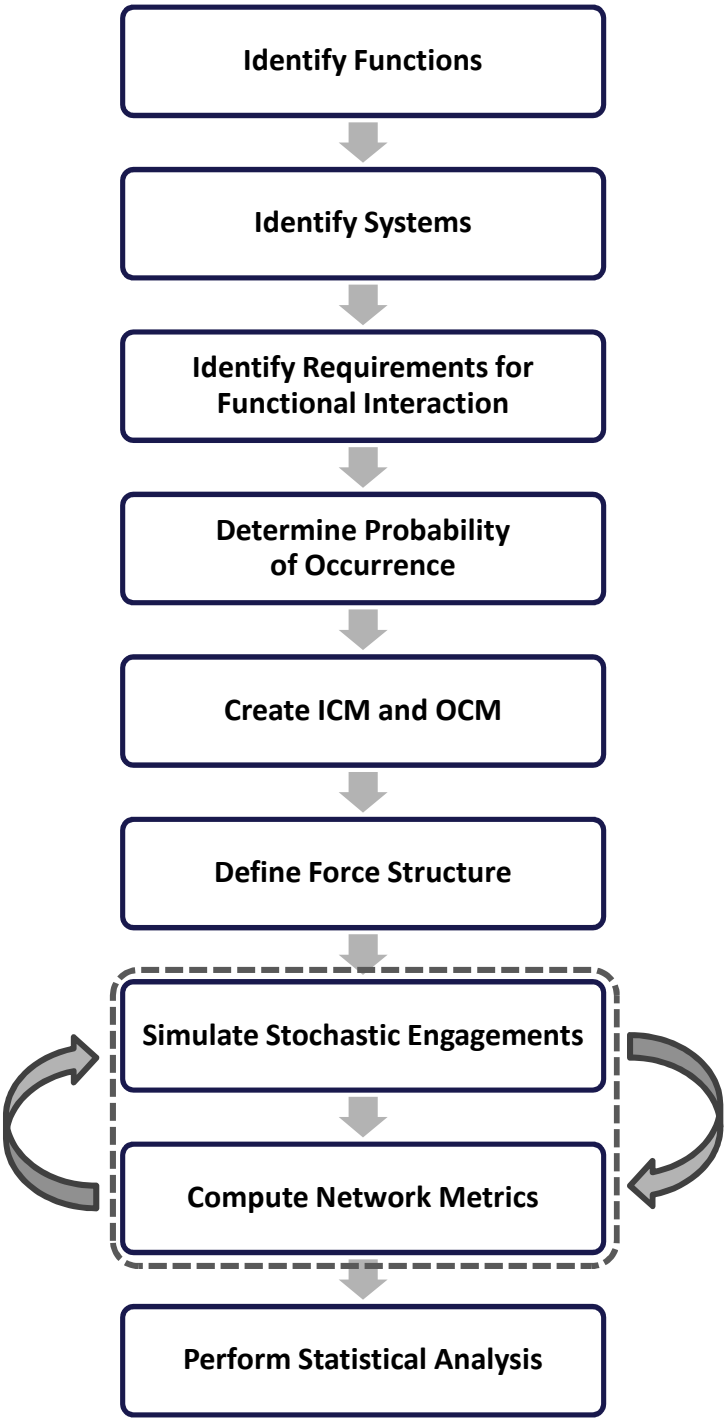

Figure 19: The Digraph Modeling for Architectures Process.

The DiMA technique is composed of the following steps:

1. **Identify Functions** Determine which functions are required to achieve the capability or capabilities of interest. Define which ones constitute the active and inert sets of functions.

2. **Identify Systems** Determine types of systems to be included in the architecture.

3. **Identify Requirements for Functional Interaction** Determine which interactions or relations must exist between systems for a functional relation to be possible.

4. **Determine Probability of Occurrence** Determine probability of occurrence of interactions between the system types.

5. **Create the In-degree Constraint Matrix and Out-degree Constraint Matrix** Specify the maximum number of inputs and outputs each system type can have for each function. These matrices tend to be sparse.

6. **Define Force Structure** Specify the numbers for each type of system.

7. **Simulate Stochastic Engagements** This step consists of creating a representative Engagement Matrix from the Engagement Generation Matrix.

8. **Compute Network Metrics** In this step the stochastic engagements can be analyzed to compute metrics from the networks (or graphs) for each of the different repetitions.

9. **Perform Statistical Analysis** The repetitions executed of the two previous steps must be analyzed to obtain statistics of the performance of the architecture.

4.1.1 Description of the DiMA process

Step 1 consists in identifying the functions and activities that are required to achieve a capability. These should not only be identified, but more importantly, they should be thoroughly understood since they form the basis of the analysis. For example, the kill chain requires basically two functions (detecting and engaging), more complex formulations, such as the Find, Fix, Track, Target, Engage, Assess (F2T2EA) Process, are simply more specific

sets of steps to achieve that capability. A networked force will still need to be able to detect and engage, but the tasks may be distributed so as to enable collaboration between the entities. For example, one entity may find a target, communicate that information, another may fix it and track it, and communicate that information, and another system target it and engage it, and a fourth assess it. The relations for the functions should be specified in such a manner that the capability achieved forms a cycle of functions amongst the systems.

Step 2 consists of identifying the systems that perform the functions required to achieve the capability. Architecture frameworks tend to include products that map functions to systems. This step can leverage those products to identify the set of systems that will take part in the completion of the capability of interest. The goal is not to identify the systems per se, but the types of systems. DiMA does not need to know the small differences between one F-16 and another, but it needs to know the differences between the different types of agents, e.g., F-16 and F-22.

Step 3 This step consists in identifying the relations that must exist in order for a functional relation to exist. For example, the function may be detect, but in order for a sensor aircraft to detect a target, it does not only need to receive the signal from the target being detected, but also be able to control its sensor suite. It may also need to be able to receive coordinate pairs from a C2 unit, the combination of requirements is infinite. The analyst needs to understand the functions that compose the capability, and he/she needs to understand the intent of the model. What is to be measured, what can be disregarded. This step serves as a check in that the analyst can study the products being created in this step and determine whether or not the analysis will capture the required behaviors. For the method to work, when a capability cycle is met, a system must be able to traverse the graph and reach back to itself through the other systems related through the functions. It may be possible that the system may reach itself even when the capability is not met, these will be referred to as *inert cycles* and will be discussed in more detail further ahead.

Step 4 will specify the probability of occurrence of an interaction between any two entities (either the existence of a needline, or a functional relationship). These depend on the systems' performance, the environment, and the way they are employed. The SV-7

(Systems Performance Parameters Matrix) can be used to aid in the definition of these probabilities, but input from experts is required to vet the values. If the probabilities are not well known, or cannot be defined properly, there are two options. The first consists in conducting a sensitivity analysis to identify which variables have the largest impact on the behavior of the system, and therefore, should be known with more certainty. The second option consists in assigning distributions to the uncertain probabilities and conduct a Monte Carlo simulation to capture the uncertainty.

Step 5 will define the In-degree Constraint Matrix (ICM) and Out-degree Constraint Matrix (OCM) for each asset and each function. In general these constraints are sparse and therefore quick to define. They can be used to reflect an individual system's Measure of Performance (MOP), e.g., in the case of a Unmanned Combat Air Vehicle (UCAV) architecture study, the number of missiles in the UCAV constrains how many targets they can engage. Therefore, for the function *engage* for the UCAV, the value in the OCM should be set to the number of missiles it will have. This allows for trades to be performed at the system level, and study their impact at the architecture level, e.g., how is the architecture's overall capability impacted when we move from a 2 hellfire UCAV (MQ-1 Predator) to a 6 hellfire UCAV (MQ-9 Reaper)? Communication and sensor technologies can also be tested by stipulating how many units can a single entity communicate with or detect and track at any one time. The ICM and OCM are of size number-of-functions$\times$number-of-node-types.

Step 6, the last step in the information elicitation portion of DiMA, defines the quantity of each type of asset in the theater of operations. Often referred to as the force structure, this can be a crucial parameter to include in the tradeoff of the architecture. This step should not only include the friendly force structure, but also the opponent's(s'). The product to create in this step is the Force Structure Matrix (FSM), which includes for each type of system, the side it belongs to and the quantity present in the theater of operations.

Step 7, the first in the analysis portion of DiMA, consists in creating the engagement matrices. For this purpose, the data elicited in the six previous steps must be synthesized into a useful whole. Steps 1 through 4 will enable the creation of the EGM, which together with the force structure and the ICM and OCM, will enable the simulation and analysis of

the architecture. The EGM is a $N \times N \times M$ matrix, where N is the number of systems involved in the analysis, and M is the number of functions. Figure 20 depicts the generic form of the EGM. In essence, this matrix contains the probability that a functional relation exists between any of the systems. It is directed (for undirected relations, the entry should be symmetric), it contains numerical data ranging between zero and one, and it is generally sparse. In the example given in Figure 20, there is a 10% probability that System N (S_N) interacts with System 2 through Function 2 (F_2) and a 15% chance with System 3 through Function 1. In graph theory, this representation of graphs is referred to as a *layered graph.*[140] A functional cycle can then be tested to see if a certain capability has been achieved, since for a capability to be achieved, the entities involved must be related cyclically through their functional relations. The activities (and their supporting functions) that are necessary to achieve the desired capability serve as the required path for traversing this layered graph. If the probabilities can be assumed to be independent (this makes the analysis simpler, and if the sparsity of the graphs to be generated is sufficiently high, it is an acceptable assumption), then the simulation of possible theater-level engagements can be done by expanding the EGM by the force structure and the different Engagement Matrices analyzed.

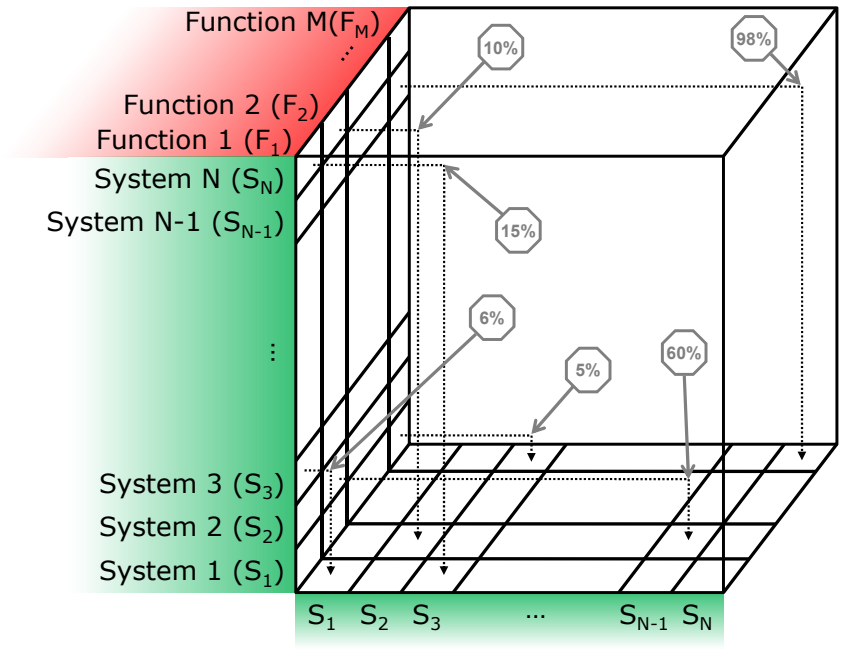

Figure 20: Generic form of the Engagement Generation Matrix.

Algorithm 4.1 DiMA Stochastic Simulations

Require: $EGM, CFM, F, R.$

Ensure: $CNE, NodeCyclicity.$

for $r = 1$ to R **do**
  for $f = 1$ to Number of Functions in CFM **do**
    for $i = 1$ to $\sum F$ **do**
      for $j = 1$ to $\sum F$ **do**
        if Random$[0, 1] < EGM(i, j, f)$ **then**
          $EM(i, j, f) \leftarrow 1$
        else
          $EM(i, j, f) \leftarrow 0$
        end if
      end for
    end for
  end for
  for $s = 1$ to Number of Sides **do**
    for $c = 1$ to Number of Capabilities in CFM **do**
      $CM_{Active} \leftarrow \sum_{f=f \in c} EM$
      $CM_{Inert} \leftarrow \sum_{f=f_i \in c} EM$
      $CNE_A(s, c, r) \leftarrow \frac{\max(\text{eig}(CM_{Active}))}{|u|}$
      $CNE_I(s, c, r) \leftarrow \frac{\max(\text{eig}(CM_{Inert}))}{|u|}$
      $NodeCyclicity(:, s, c, r) \leftarrow \text{diag}(CM^{|u_s|+1})$
    end for
  end for
end for
return Average over $r, Coefficient of Networked Effects(CNE), NodeCyclicity.$

Step 8 consists of computing a series of network metrics for each Engagement Matrix (EM). Several of these engagement matrices are generated to statistically study the different

possibilities that may occur in reality. Since the focus is in studying the ability of an architecture to complete a capability, where the capability is described as a cycle of functions through the systems that compose the architecture, the metrics employed focus on measuring the cyclicity in the engagement matrices. Nonetheless, it is important to emphasize that an approach similar to this one can leverage the same steps described above but study a different set of network metrics to obtain other insights into the behavior of a networked system. The network metrics used include: The largest value eigenvalue of the adjacency matrix as a measure of the *functional cyclicity* ($\lambda_1^{\mathcal{A}}$), and its normalized value (CNE), its associated eigenvector (PFE) as a function of the node centrality, the node cyclicity for each stochastic engagement, and the Fiedler Vector ($\mathbf{x}^{\lambda_2^{\mathcal{L}}}$) as a measure of the network's ability to reach steady state.

A MATLAB function was created to execute the stochastic simulation portion of DiMA. This function simulates likely stochastic engagements based on the force structure and EGM, and for the different capability cycles required, it computes the metrics of interest. The pseudo-code is presented in Algorithm 4.1 and the MATLAB function in Appendix F.4.1. This algorithm requires as an input the Engagement Generation Matrix (EGM), the functional cycles requires to achieve each capability (Capability Requirements), the force structure (Force), the number of repetitions to execute. EGM is represented as a $n \times n \times f$ matrix, where n is the number of types of systems, and f is the number of functions. CFM is the Capability-Function Matrix, a $c \times f$ matrix specifying for each capability c which of the f functions are required to achieve it. Force is a $n \times 1$ vector containing the force structure of the architecture(s). R is a scalar with the number of desired stochastic repetitions to be executed. The function is presented in Appendix F.4.1. The function automates the creation of the Engagement Matrices EM, it conducts a MCS on them by executing a large number of random EGMs, and compiles the means of the PFE, Fiedler Vector (FV), and if desired, successive adjacency matrix multiplications to compute the cyclicity of the network and each node. These metrics measure the cyclicity of the graph in slightly different ways, and can be used to understand the behavior of a networked system as described in Section 4.1.3. These are just two approaches to measuring the cyclicity of a digraph, for more on

this discussion the reader is referred to [196, 267]. As stated by Popescu et al. [267], there are a multitude of ways to measure the cyclicity of a graph. The most straight forward approach consists in adding the identity matrix to the adjacency matrix and successively multiplying the sum of the two to produce a reachability matrix, where the (i, j) entry of the matrix is the number of paths a node can have to reach any other node. For more information, please refer to Appendix E.1. A more elegant approach consists in studying the spectrum of the graph, in particular, the left and right PFEs and the FV. Multiplying the left and right PFEs will give non-zero values to the nodes that are in the major cycles of the graph. The FV can also be used for this purpose. The problem lies in that the node may be within a different cycle, which might still contribute to the capability of the architecture, but will receive a value of zero by the left and right PFE multiplication. The successive matrix multiplication is an alternative approach, the problem with this method is that the value of the entries of the matrix quickly increases and it can be prohibitive to calculate. A cycle in a graph can be as long as the total number of nodes, this means that for a 100 node matrix, a 100-by-100 matrix must be multiplied 100 times, this operation requires $O(n^3)$ flops. MATLAB uses LAPACK to compute eigenvalues and eigenvectors, which requires $O(n^2)$ flops, this is a significant reduction for large matrices, which is what DiMA is meant to support. At the same time, if information is known about the maximum number of elements a cycle can have, e.g., for the Joint Unmanned Combat Air System (J-UCAS) example, even though there might be a large number of targets, the cycle cannot be greater than the total number of blue units plus one (for the target in the cycle). It is not possible in this formulation for a cycle to exist within the red force since they do not perform functions that can affect each other. Therefore, the matrix multiplication can be curbed to the maximum cycle size.

The eigenvalues also offer insight into the cyclicity and structure of the graph. As mentioned in the appendix, the number of zero eigenvalues of the Laplacian matrix indicates the number of components in the graph. If only one zero value is identified, all elements of the graph are connected. If there are two or more zero eigenvalues for the Laplacian, their associated eigenvectors will contain non-zero values for the nodes that are part of

that component. Therefore, if the graph is not connected, and the user is interested in identifying the nodes in the two components, the associated eigenvectors to the zero-value eigenvalues will offer than information.

Step 9 consists in obtaining statistics from the metrics obtained in **Step 8**. The statistical reductions can simply be the mean over the repetitions executed, but **Step 8** produces a wealth of data which can offer more insight than just a simple comparison. Standard deviation of CNE can give an indication of the variability in capability a given architecture will display.

4.1.2 Leveraging DoDAF with DiMA

The following is a summary of the DoDAF products that can be leveraged by DiMA to aid in the analysis of military architectures.

OV-2 Operational Node Connectivity Description Specifies the operational nodes and what communications they require (Step 3).

OV-5 Operational Activity Model Specifies the activities that must be completed and their relationships (Step 1).

SV-4a Systems Functionality Description Specifies which systems perform which functions (Step 2).

SV-5a Operational Activity to Systems Function Which functions enable which activities (Step 1).

SV-7 Systems Performance Parameters Matrix (Optional) To aid in the definition of the probabilities that two entities interact functionally.

Unlike other proposals to make DoDAF executable, DiMA incorporates standard DoDAF products without the need to modify them or extend them. The amount of work required from the time the DoDAF products are defined is minimal when compared to formulations that require the creation of Petri Nets, or Discrete Event Simulations. For these reasons, it is considered that DiMA offers a capability currently insistent in the open literature.

4.1.3 Example Analysis using DiMA

A hunter-killer Unmanned Aerial Vehicle (UAV) architecture selection problem will be used to illustrate the steps and demonstrate how the technique can be applied. The capability of interest is the ability to eliminate enemy targets. The capability of interest is to eliminate enemy targets. In the example, there are 7 entities of interest, C4ISR stations, hunter and killer UCSs, hunter and killer UAVs, communication satellites, and targets. The assets perform 4 functions: detection, communication, control, and engagement. Figure 21 describes how the asset types interact through the different functions they perform. The problem of interest is to study the level of interoperability[1] required to achieve the capability of interest. In order for the blue side to achieve the capability of eliminating enemy targets, five activities must be completed (these data can be obtained from the OV-5, SV-4a, and SV-5a):

1. The sensor aircraft (hunter UAVs) must be able to detect the target,
2. that UAV must be able to communicate that information to its control station,
3. that station must be able to communicate the information to a killer station, (this can be done directly, or through a C4ISR station, another UAV Control Station (UCS), or a satellite uplink if the interoperability level of the architecture is high enough)
4. that killer station must be able to direct the killer UAV to the target,
5. and the UAV must be able to engage the target.

These data represent the results of completing **Step 1** and **Step 2**. In this simple demonstration, the enemy is not reactive and therefore cannot affect the friendly systems, it is not of interest to test the enemy's capability either.

Step 3 must determine the relations required for a functional interaction to occur between any two given systems. There are multiple communication channels available

[1] A measure of how well the different entities interact, low levels indicate that the communications links are sparse and a unit can only interact with a few others, higher levels indicate that most units can communication with most other units.

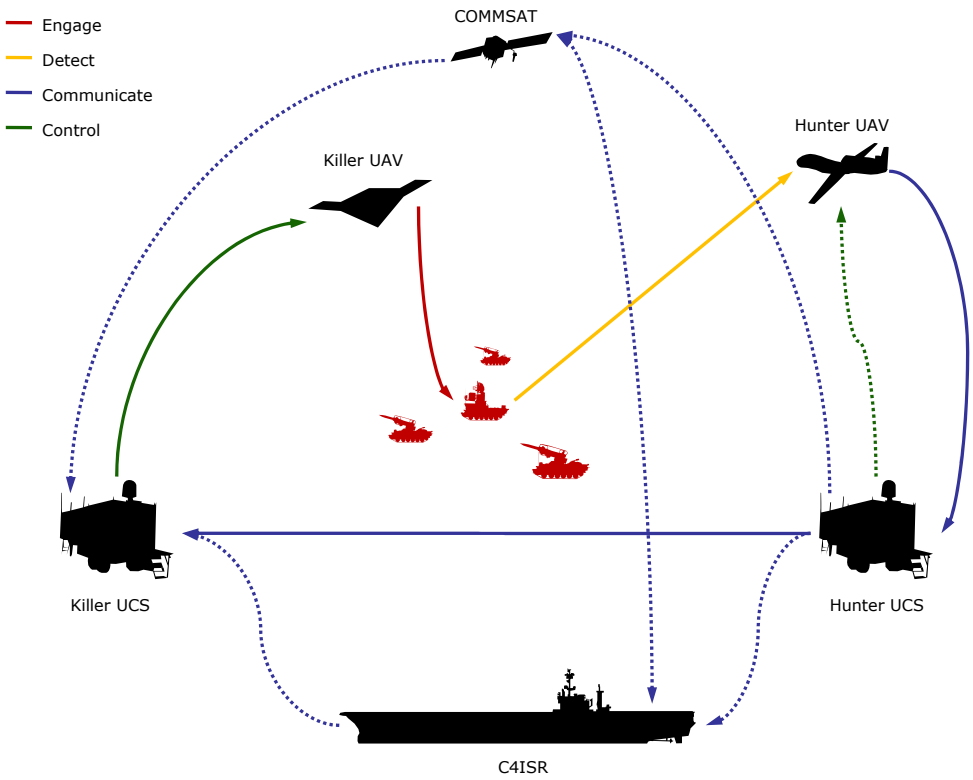

Figure 21: UAV architecture and functional relations.

to the systems, there is a (1) system status link, a (2) payload control link, a (3) data link, and a (4) system command link. The number of operational channels required for an activity to take place, e.g., control, detection, etc., varies depending on the level of autonomy of the UAVs. A fully autonomous UAV–e.g., a hunter UAV with advanced image recognition algorithms to identify and track enemy units and control its own camera– does not require a payload control communication link to exist between itself and the UCS. The different architecture alternatives dictate which links are allowed, and the system technologies implemented determine which links are required to complete an activity. For example, an Inter-Operability Level (IOL) of 0 does not allow satellite communications, so there are no links between the satellite assets and the rest of the units, and non-autonomous UAVs require system-control link, payload-control and data link in order to identify and track a target, and engage it. In the DoDAF nomenclature, the nodes of interest are operational nodes, and the information required are needlines. Figure 22 represents the example provided in Figure 4-3 of DoDAF 1.5 vol. 2.[107] This figure describes the elements of an OV-2: Operational Node Connectivity Description view.

Operational Nodes An operational node is an element of the operational architecture that produces, consumes, or processes information. What constitutes an operational node can vary among architectures, including, but not limited to, representing an operational/human role (e.g., Air Operations Commander), an organization (e.g., Office of the Secretary of Defense (OSD)) or organization type, i.e., a logical or functional grouping (e.g., Logistics Node, Intelligence Node), and so on. The operational node will also vary depending on the level of detail addressed by the architecture effort.

Needlines and Information Exchanges A needline documents the requirement to exchange information between nodes. The needline does not indicate how the information transfer is implemented. For example, if information produced at node A is simply routed through node B and is used at node C, then node B would not be shown on the OV-2 diagram the needline would go from node A to node C.

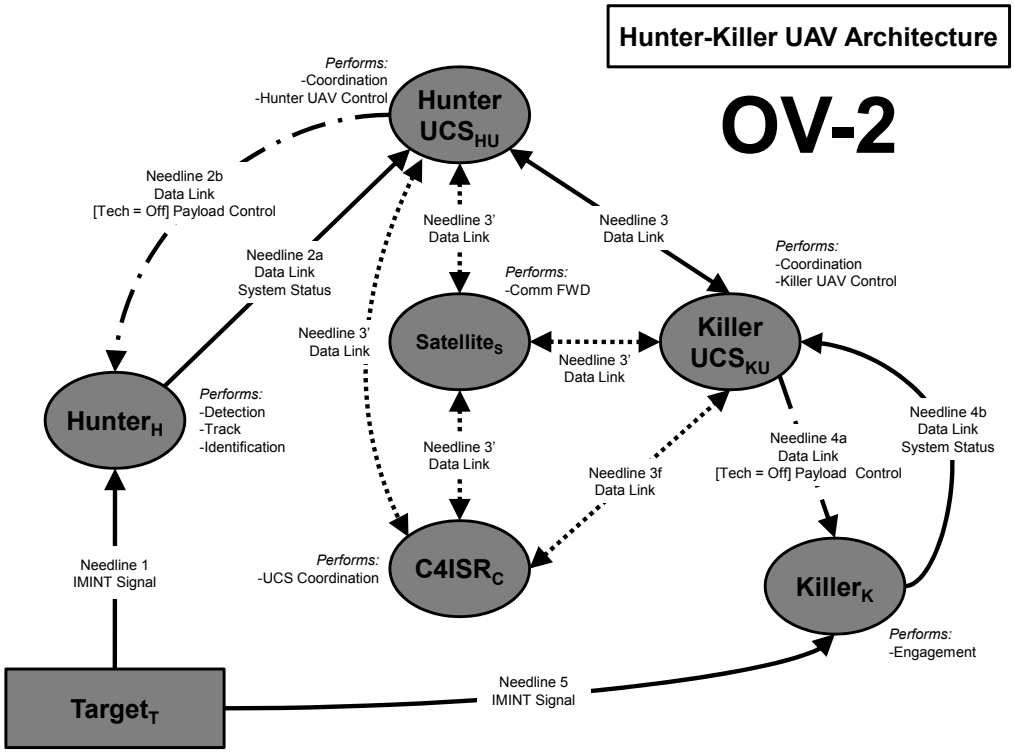

Figure 22: OV-2 Example.

The concept of the needline may not seem intuitive, but integrates well with DiMA because the physical communication links must exist between the needline nodes (either directly or routed through other nodes) but the function specifies the need for a connection between the two entities in order for the node to be able to fulfil the required function. For example, needline 3 (the requirement that the Killer UCS receives target information from the hunter UCS and the hunter UCS receives an acknowledgement from the killer UCS) can be rerouted through the Satellite, or through the C4ISR Station, or a combination of the two.

A 12,048 case Design of Experiments (DOE)–a hybrid fractional factorial (Central Composite Design (CCD)), space filling (Latin Hypercube) design–was executed and analyzed to obtain insight into which parameters produced the largest effect on the architecture's ability to complete the desired capability. Table 9 contains the variables that were included in the DOE, along with their symbol and ranges. The execution the DOE produced a variety of results, including sensitivities and capability estimations. Figure 23 contains some of the results from the Analysis of Variance (ANOVA) of the DiMA outputs of the DOE.

Figure 23 contains the Pareto plots for four different factors of interest, Active CNE, Ratio of CNE Active and CNE Passive, the Net CNE, and the Cyclicity of the targets (an indication of how many were successfully engaged). For the Active CNE, 90% of its variability is a function (within the ranges chosen for the DOE) by (1) the probability of the killer UAV engaging a target, (2) the number of hunters, (3) the number of Killer Control Stations, (3) the probability of detection of a target by a hunter, (4) probability of the control linking existing, and finally (5) the probability of the communication link existing. The autonomous technology and the number of missiles on the killers seems to be irrelevant to the overall architecture's capability. Note that the active CNE includes the latent structure, that even though it may contain a large number of latent cycles, those do not truly affect the architecture's ability to complete the desired capability cycle.

The CNE ratio, which can be considered to be related to the efficiency of the architecture since it compares the total cyclicity to the inert cyclicity (that which is there to support the capability but does not produce a capability of its own), is only affected by four parameters,

Table 9: Ranges for Control Variables.

Variable	**Short**	**Min**	**Max**
Maximum Targets Killed per Killer	MTK	2	6
Autonomous Tracking and Engagement	TECH	0 (OFF)	1 (ON)
Inter-Operability Level	IOL	0	4
Probability of Detection	p_D	5%	45%
Probability of Engagement	p_E	5%	65%
Probability of Communication Link Operational	p_{Comm}	90%	99%
Probability of Control Link Operational	p_{Ctrl}	90%	99%
Number of C4ISR Stations	nC4ISR	1	5
Number of Hunter UAV Control Stations	nUCSH	1	5
Number of Killer UAV Control Stations	nUCSK	1	5
Number of Hunter UAVs	nH	5	30
Number of Killer UAVs	nK	5	30
Number of Communication Satellites	nS	1	3
Number of Targets	nT	15	55

the number of Control Stations, and the technology. This is an interesting and logical result because the control stations are the ones that contain most of the latent cycles, so the fewer there are the smaller latent structure, and therefore, the most efficient the architecture can be. The autonomy technology is also a reasonable result since with it more cycles in the latent structure exist. The unexpected result is that the parameters that affect the active structure were not as critical. This may be an indication that due to the ranges chosen, the control stations dominated, while within their more controlled ranges, the probabilities of detection and engagement had a lesser effect.

The net CNE is the parameter that may be of most interest as to how well the capability is being performed. This once again provided insightful but unexpected results. For this metric, 95% of the variability could be explained by (1) probability of detection, (2) the number of killers, (3) the number of killer UAV control stations, (4) the amount of ammunition per killer, (5) the number of targets, (6) the probability of engagement, (7) the number of satellites, (8) the number of hunter UAV control stations, and finally (9) the number of hunter UAVs. The IOL was conspicuously unimportant, a result that is most unexpected, and the autonomous technology seemed to have the least effect. The reader is reminded that these results are only valid for the ranges chosen, nonetheless, it is an example of a study that a decision maker could very quickly execute and obtain insight as to what parameters of the executed should be pursued with more interest. It also points to the fact that having a high probability of detection is extremely critical, as it is to have killer UAVs in sufficient numbers.

Finally, the cyclicity for the targets is presented in the lower right quadrant of Figure 23. This metric is the average number of targets that are within an active cycle of the blue force (for this case the cycle has been constrained to those of at most length 10) and it can be considered to be a measure of how many targets were successfully engaged. This parameter was most affected by (1) the number of targets, (2) the number of killer UAVs, (3) the probability of detection, (4) the number of hunter UAVs, (5) the maximum number of targets a killer UAV can engage, and (6) the probability of engagement. It is logical to assume that the number of targets would have a great effect on how many targets were

engaged, especially considering the large range given to that parameter and the fact that this measure is not normalized by the number of targets included, i.e., it is not percent of targets engaged, its simply targets engaged.

When first faced with these results, it seems like a sensible assumption that the cyclicity of the targets and the net CNE should be dependent on the same parameters since it seems that they are measuring the same thing, the ability of blue to kill red targets. But as it will be made evident shortly, this is not truly the case. There are a some significant differences that make the different metrics capture different characteristics of the architecture. The cyclicity is computed as the number targets that fall within a kill cycle of blue, the net CNE on the other hand is the difference between the active and inert CNEs. The net CNE then can be influenced by both the active and passive portions of the architecture, so in a way, it acts as a measure of the efficiency, there can be very few (or even no) inert cycles in a given configuration, but a number of active ones. In that case the CNE will be large, or there can be a large number of active and inert cycles, making the net CNE much smaller in magnitude. The net CNE is in a way a non-normalized capability efficiency, much like the CNE ratio is a normalized version of that efficiency. It is not difficult to understand then that efficiency is not the same as efficacy, net CNE is a representation of efficiency, while the cyclicity of the targets is a measure of the efficacy of the architecture. This is a good example of why it is important to look at as many complementary metrics as it is possible, since the insight when be derived when these are compared, not when they are studied in isolation.

This analysis is not meant to be a rigorous study, or a test of the technique, it is simply an example to illustrate how DiMA can be used. For more information on the testing the reader is referred to the hypothesis testing section.

4.1.4 DiMA and the OODA Loop

When first introduced to DiMA–and this example in particular–readers may think that the process shares great commonality with Col. John R. Boyd's OODA loop. The reader is referred to Figure 24 for a visual representation of the OODA loop. Strictly speaking, Boyd's

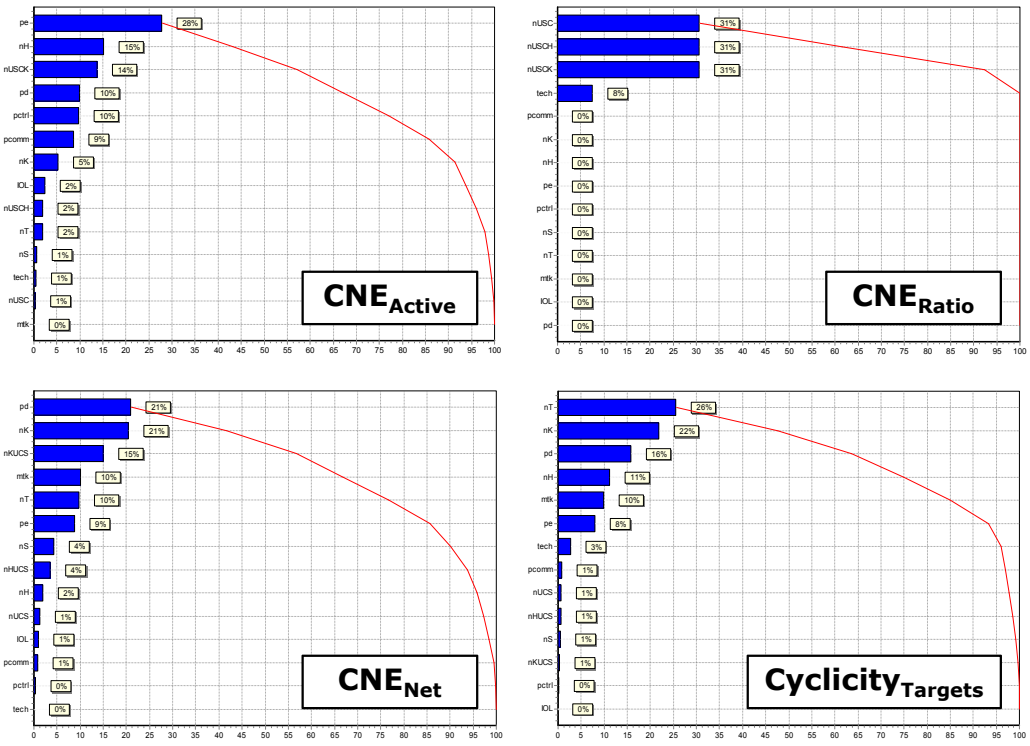

Figure 23: Results from the J-UCAS implementation of DiMA.

formulation (as originally conceived, e.g., the common example is that of a fighter pilot iterating through the OODA loop during a dogfight) applies to the internal workings of decision makers, and is therefore not conceived as a distributed process, while DiMA is meant to aid in the modeling of a distributed architecture's ability to fulfill a capability. For the generalized formulation of the OODA loop, where the different functions can be performed by distributed entities, the example shown mirrors the OODA loop in the following manner. The sensor craft *observes* and transmits the information to a decision maker. The decision maker orients itself by analyzing possible courses of actions, e.g., asking its subordinates to perform tracking and identification tasks, or send a killer UAV to eliminate the detected entity, but to do this the decision maker needs to assess other information, as for example, what is the status of its subordinates. The decision portion is the selection of the course of action, and finally, the action is the execution of that decision, which in turn will affect the environment–meant as in the general case, where it is not only the physical environment, but the other entities and the enemy as well–of the decision maker. Note that even in the distributed case, the OODA loop formulation is focused on the internal processes of the entities (in this case the decision maker), DiMA on the other hand, attempts to leverage the complexities of distributed system architectures, by focusing on their interactions, and not necessarily on their internal complexities. It attempts to capture the complexity of their behavior by studying their structural (relational) complexity.

4.2 Experiments to Test DiMA

The first experiment must check the validity of the approach for modeling the complex systems using DiMA. In this experiment, complex ABMs will be used to obtain measures of effectiveness, as well as the parameters that are inputs to the simpler digraph models to ensure that when given correct data, DiMA can produce sensible results that correlate with the ABM&S. A series of increasingly demanding tests are performed to study the validity of the approach. These are listed below.

1. Time step EGM analysis of the PFE's ability to identify the completion of a capability cycle.

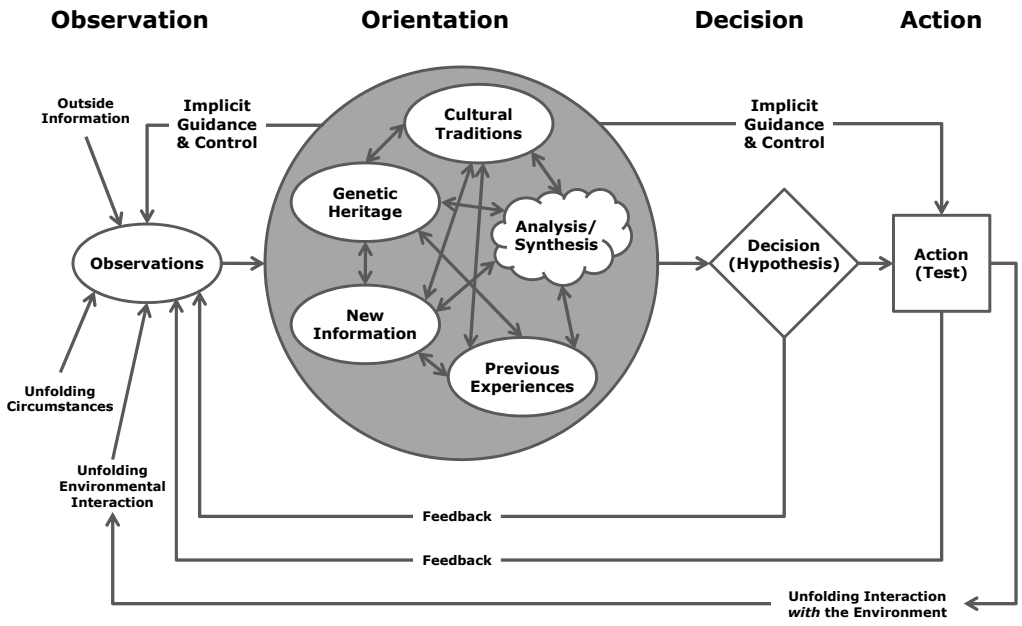

Figure 24: Boyd's OODA Loop.[275]

2. Time averaged EGM analysis of the PFE's ability to predict the outcome of a complex agent-based simulation.

3. Variability in the time-averaged EGM to ensure that for a certain force structure, measures of performance, and scenario, one unique EGM is produced.

For DiMA to fail, there should be low correlation between the its outputs and those from the agent-based models.

4.3 Models to Test DiMA

Three ABM&S models were employed to test the validity of the digraph modeling technique to assess the capability of an architecture. These models were programmed in NetLogo 4.0 [340] due to its ease of use and free availability. NetLogo can be readily integrated with Wolfram Research Mathematica®and MathWorks™MATLAB®, or executed in batch mode with Phoenix Integration's ModelCenter®to study large numbers of simulations, or executed from both of these integration schemes were used when executing the following models.

4.3.1 AirWar v13

AirWar v13 is the first of the agent-based models that display "complex" behavior used to test DiMA, and consequently, Hypothesis A. AirWar v13 is an air and ground engagement model between blue air forces, and red air and ground forces. AirWar contains 7 types of units, presented in Figure 25, four blue and three red. The blue units include E-3As, F-22s, F-35s and RQ-4As. The red units include MiG-29s, radars and SAM launchers. The E-3As Sentries can detect and track enemy MiG-29s from long ranges, as well as vector F-22s and F-35s to them. The RQ-4 Global Hawks can detect ground targets and communicate those detections to the E-3As which can then vector F-35s to them. F-22s and F-35s have short range detection capability, F-22s can only engage air targets, while F-35s can engage both. F-22s have 6 Air-to-Air Missile (AAM)s, while the F-35s have 2 AAMs and 4 Air-to-Surface Missile (ASM)s. The enemy forces are composed of Surface-to-Air Missile (SAM) batteries, each with a number of SAM launchers controlled by a Straight Flush Radar (SFR), and

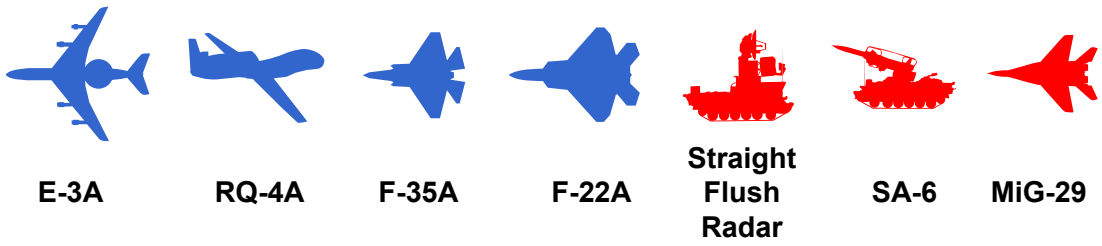

Figure 25: Units in AirWar v13.

MiG-29s. The MiGs are vectored by the radars to incoming blue aircraft in a similar fashion as the E-3As vector F-22s and F-35s. While waiting for vectors, MiGs patrol horizontal corridors in the red area of the theater of operations. All fighters and attack aircraft fly in wings. The F-22s fly in wings of 2, the F-35s in wings of 3, and the MiG-29s in wings of 4 aircraft. Each wing has a designated wingleader which is replaced by the highest ranking wingmen when that wingleader is shot down or exits the simulation.[2] Communications between the C2 units (the E-3As for the blues and the SFR for the reds) to the fighter and attack wings is done through the wingleader, who in turn communicates with his/her wingmen. The communications range is finite and it can be set to a predefined value.

The SAM batteries require that the SFR track a target for a certain amount of time, then communicate with a SAM launcher that has SAMs and that it has the target in its field of view. The SAM launcher can then shoot the SAM which must track and detonate within a certain distance from the aircraft. The probability of kill is based on the distance of detonation. SAM launchers have 3 missiles and as the rest of the units, they cannot be resupplied. Figure and attack aircraft launch missiles which must also detonate within a kill radius to eliminate the target.

4.3.2 Enhanced Distributed Networked Operations Simulation

Enhanced Distributed Networked Operations Simulator (eDNOsim) is based on J.R. Cares' DNOsim, a model created to capture the complexities of networked operations and serve as a

[2]When attack aircraft and fighters run out of weapons, they proceed to an egress point from which they exit the simulation.

tool to better understand them. eDNOsim was extended to include complex decision making processes and it adapted the roles of the different types of assets. In the original Distributed Networked Operations (DNO) formulation, the agents can only belong to one of four possible categories: sensors (S), deciders (D), influencers (I), or targets (T). Targets were included in the original formulation because it may be possible that there are assets that have military value but are neither sensors, deciders, nor influencers. In this enhanced version, the targets were transformed into tankers (or in more general terms logistic support units), so they provide a function to the other warfighters, and are not simply idle bystanders.

The sensors are the only units that can detect the presence of enemy units. The detection is probabilistic and based on the Radar Cross Section (RCS) of the enemy unit, the range to it, and the signal-to-noise required by the sensor. The radar equation was used as the basis for determining detection. Equation 6 presents the signal-to-noise ratio created by a normalized RCS (σ) with a baseline of 5, and range to target (R) by a nominal detection range (R_0). The sensors must then communicate the detection to the deciders which can then begin to track and identify. Tracking is determined when continuous detection by the sensors and communication to the deciders exceeds a certain amount of time.[3] An additional amount of time of continuous detection and communication to the decider allows deciders to identify the target.

$$S/N = \left(\frac{\sigma}{5}\right)^8 \left(\frac{R_0}{R}\right)^4 \tag{6}$$

Deciders can vector influencers towards a target as soon as the target has been tracked. Once the target is identified, the decider can prioritize between all the possible targets identified to vector the available influencers to the closest target. When multiple influencers are vectored to the same target by multiple deciders, if these deciders can communicate, they will de-conflict the vectors to ensure that only the most suitable influencer is vectored. In the communication architecture implemented, deciders are the only units that can communicate with other units and the only type of unit which can receive communications.

[3]There can be multiple sensors detecting the same target and communicating it to the decider.

Influencers have a limited number of weapons, and a minimum distance they must approach their vectored targets before they can engage. Furthermore, influencers must engage continuously for a certain amount of time before they can fire. Influencers with no weapons cannot engage and are not vectored by the deciders.

When a unit reaches "Bingo" fuel, it requests a refueler. Deciders are the only ones that can request directly to the tanker, all others must request via a decider. To refuel, an asset must approach the refueler and stay behind it for a certain amount of time. Tankers have limited amount of supply fuel and a finite rate at which they can deliver fuel.

Since the sensors, deciders, and influencers are all necessary for the force to be able to eliminate enemy units, the loss of all units of a type renders the force completely ineffective. Once all the units of a type are lost, the simulation stops and records it as the victory condition.

Figure 27 is a screen capture of the eDNOsim Graphical User Interface (GUI). The pre-setup controls must be set before the model is initialized, these values set the force structure and many of the properties of the model. These values should not be changed without re-initializing the model. blue and red force controls determine the capabilities of the blue and red side and their systems. The random seed sets a specific random seed to enable repetition and ease debugging. The performance analysis controls allows the user to analyze the performance of the procedures and identify bottlenecks. The visualization controls permit the user to magnify the units, display the functional links, and plot. The state of the system can be loaded and saved using the load/save controls. The operational controls allows the user to modify the aggressiveness of the two sides and their units. The 2-D theater of operations displays the location of the assets, their heading, and if enabled, the functions they are preforming to one another. The force-level metrics present high-level information about the engagement. The first chart compares the fraction of value remaining. Value is computed as a fuel and weapons used, as well as assets lost. The units detected, tracked, and identified displays how many of each side's units are detected, tracked and identified by the opposing side. The fuel chart displays how much of fuel each asset has left on average, and the minimum of them all. The metrics by asset type displays

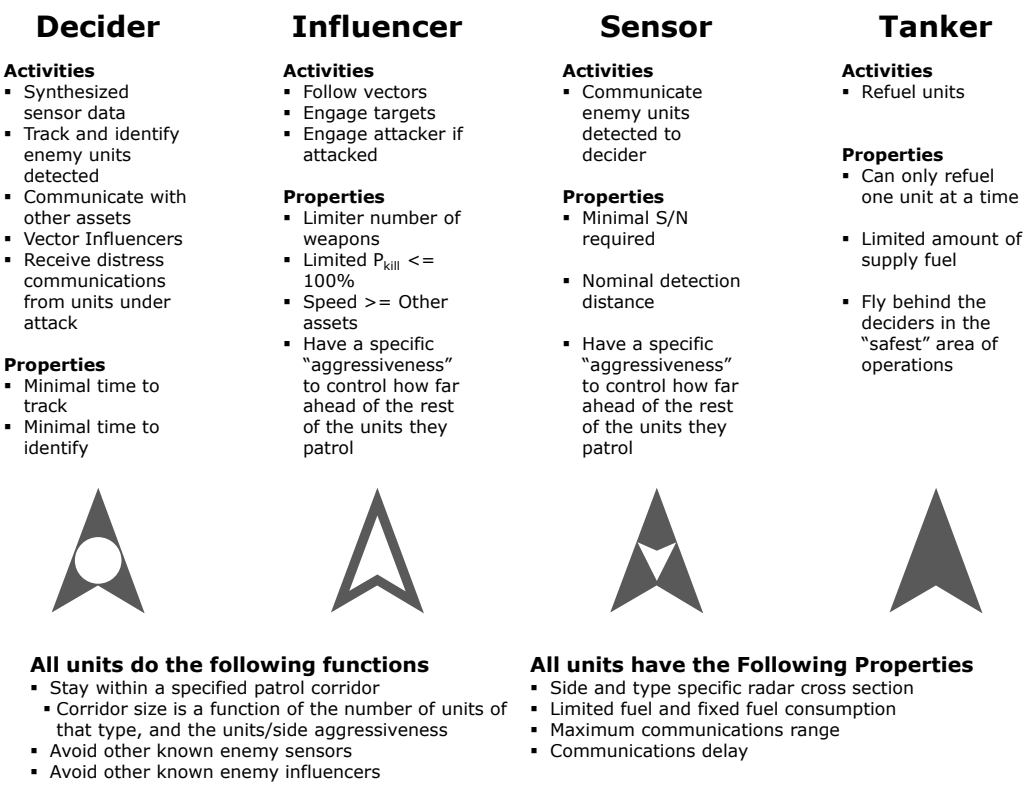

Figure 26: Units in Enhanced Distributed Networked Operations Simulator.

the fraction of assets still alive, and the fraction performing a given activity, e.g., sensing, engaging, vectoring, refueling, etc.

4.3.3 Generalized Distributed Networked Operations Simulation

The Generalized Distributed Networked Operations Simulator (gDNOsim) is another version of DNOsim where assets do not exclusively belong to any of the four categories included in DNOsim and eDNOsim, but can have properties (and follow methods) from any of them. In gDNOsim units can be targets but no supply functionality has been included in this version. The simulation is not as complex as eDNOsim but it allows for virtually infinite types of agents to be included in the simulation. This model will allow for the integrated testing of not only hypothesis A but also demonstrate how hypothesis B can be integrated with hypothesis A to create a comprehensive modeling process for complex system architectures.

The simulations using gDNOsim were characterized by two sides and 20 different types of agents. Table 10 contains the description of the 20 agent types. The first three columns are colored blue or red depending on the side the type of agent is on. The coloring on the rest of the columns is determined by how good the agent type is compared to the others for that specific property, with green being better, and red being worse. The last column, whether the agent can act as a decider or not, is colored green if it is a decider and red if it is not.

The behavior of the agents in gDNOsim is similar to that of the agents of eDNOsim, but with the additional element that any agent can be a sensor, influencer, and/or decider. Sensors still use a normalized radar equation to perform detections. Successive detections are required to track a target, but there is no identification process. Influencers have a limited amount of ammunition, have a limited shooting range, and must keep the target within a field of view to avoid loosing it. Deciders still vector influencers (unless they prefer to pursue their own targets in case they are influencers themselves). They can synthesize sensor data from sensors with whom they can communicate and they can provide assistance to units that are in distress. Smaller differences include the ability for all agents to communicate with each other or only communicate with deciders. This feature speeds the

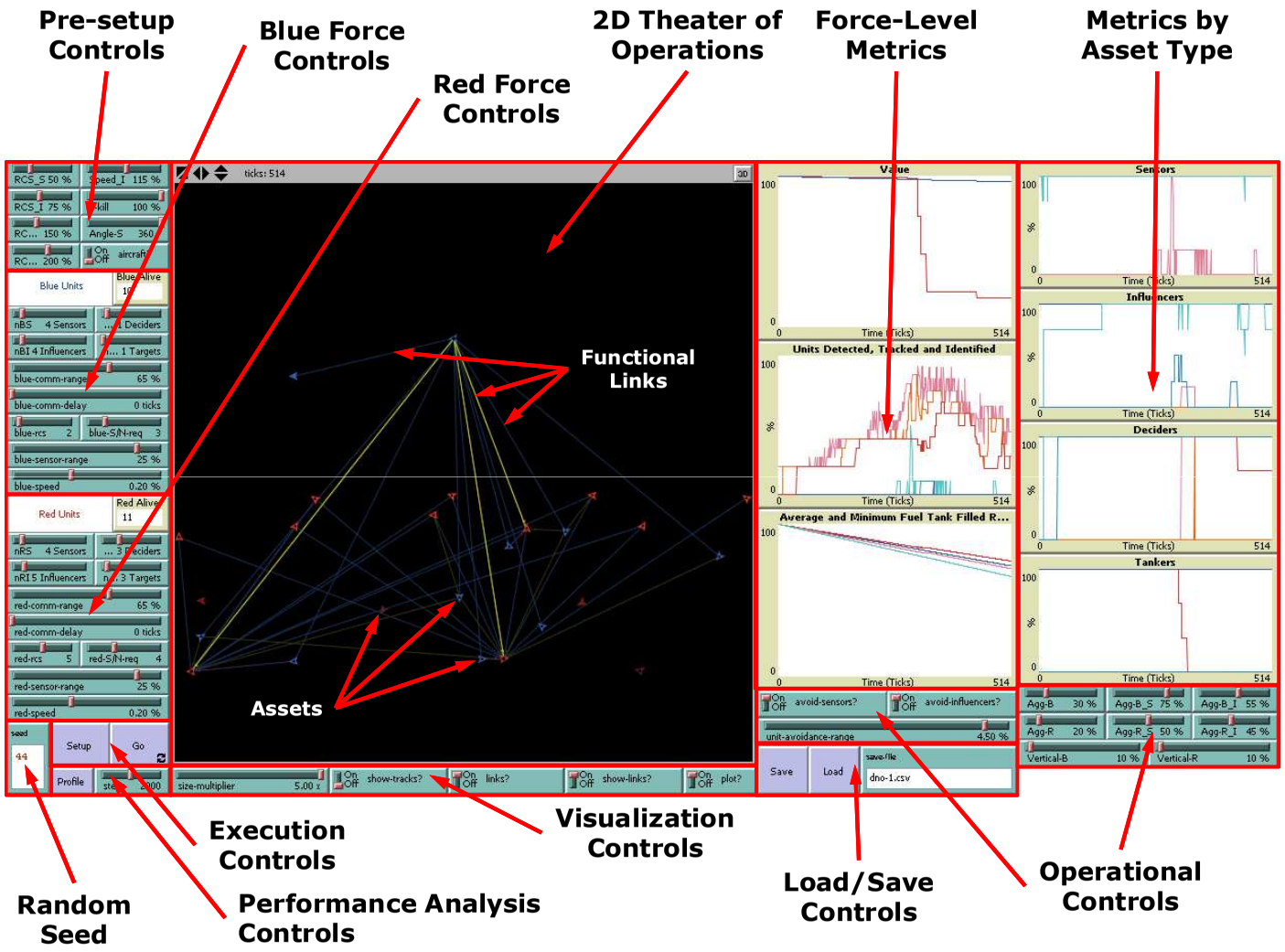

Figure 27: Enhanced Distributed Networked Operations Simulator's Graphical User Interface.

Side	NetLogo ID	N	Speed	Max-Turn	RCS	Sensor Range	Sensor Angle	S/N Required	Time-to-Track	Ammo	Shoot Range	Min Time to Lock-On	Comm Range	Decider
Blue	3	21	0.49	11.6	6.11	25.5	97	3.67	5.2	9	17.8	6.1	61.8	F
Blue	4	24	0.33	8.7	6.92	7.7	153	7.29	10.3	11	13.3	18.5	26.1	F
Blue	5	10	0.62	12.7	6.07	2.2	105	8.82	5.9	13	4.9	11.5	27.5	F
Blue	6	16	0.56	14.8	2.34	39.8	140	6.00	14.0	2	11.9	11.8	70.1	T
Blue	7	21	0.32	6.1	2.94	8.3	95	6.14	9.2	14	18.5	10.8	22.8	F
Blue	8	12	0.32	14.4	1.40	28.6	43	8.54	11.1	15	8.2	10.9	23.4	F
Blue	10	14	0.33	5.3	1.69	7.5	42	2.60	7.4	8	1.3	12.5	43.8	F
Blue	15	22	0.36	12.3	3.15	17.7	301	6.38	12.8	10	7.4	16.3	72.9	F
Blue	16	17	0.36	5.7	4.57	29.5	153	8.78	8.0	0	15.7	16.3	25.2	F
Blue	18	22	0.72	8.1	4.19	30.6	323	5.66	11.0	10	11.1	10.5	49.9	T
Blue	19	16	0.42	9.7	7.68	21.3	333	8.33	11.8	12	7.2	17.1	31.4	F
Red	1	14	0.80	7.6	6.59	12.3	46	4.99	14.2	1	2.1	17.5	10.4	F
Red	2	23	0.45	12.8	8.18	4.7	358	5.95	13.0	13	10.5	12.6	8.9	T
Red	9	20	0.73	6.4	3.88	7.2	219	2.84	14.8	1	14.1	5.4	38.1	T
Red	11	15	0.32	7.0	7.24	23.0	271	3.00	12.0	7	18.2	9.0	75.0	F
Red	12	12	0.27	11.2	1.83	6.4	182	2.10	6.2	9	14.6	3.4	20.3	F
Red	13	23	0.74	9.0	2.52	13.7	214	2.38	14.8	12	7.4	15.7	78.2	F
Red	14	16	0.75	11.4	2.42	10.9	85	3.78	7.8	7	13.8	13.6	68.6	F
Red	17	10	0.35	11.3	2.08	39.0	179	1.83	10.9	5	5.4	2.2	35.2	F
Red	20	12	0.68	9.9	2.96	10.8	139	7.65	9.1	1	7.4	15.5	42.5	F

Table 10: Unit types in the gDNOsim test for Hypothesis A.

simulation considerably, but it removes a degree of complexity that tends to produce more intricate emergent characteristics in the large-scale system architecture. The simulation also includes the option to create a fractal terrain with complex mobility characteristics that impede the movement of the assets. The final feature that gDNOsim includes that eDNOsim does not, is the idea of aggressiveness. When the simulation starts, both sides have equal aggressiveness (and unless they are pursuing a target) will remain in their respective sectors. Whenever a unit is killed, the killer will communicate that to the other agents it can interact with, and their aggressiveness will increase, meaning that they will advance and opt to claim more territory as their own. Similarly, the agents with whom the killed agent could communicate at the time of its demise, will reduce their aggressiveness and retreat. Due to the fact that not all agents can communicate with all other agents, the aggressiveness of all agents on one side is not the same. The distribution of aggressiveness nonetheless, is an indicator for which side is enjoying the advantage.

4.4 Time Step Analysis of an ABS with the DiMA Technique

This experiment will test that there is in fact a 1-to-1 relation between a capability (in this case enemy force neutralization) and a functional cycle (detect, communication, vector, engage), and that the determination of an architecture's capability accomplishment can be done with the DiMA technique by analyzing the cyclicity of the engagement matrices created by DiMA. The functional cycles in the network will be measured by the largest eigenvalue of the adjacency matrix ($\lambda_1^{\mathcal{A}}$) as described in Section 4.1 and Appendix E.

In this experiment, the AirWar v13 simulation was stepped through time by Mathematica using the NetLogo Mathematica Link. Meaning that, at every time step, Mathematica obtained the communication, orders, detection and engagement networks in the NetLogo model and used those to create a blue and red force "capability" network, and compute the spectrum of the graphs. The capability networks were created by overlaying the communication and order links between the friendlies, the tracking links from the enemy units to the friendly units, and the engagement links from the friendly units to the enemy units, where friendly and enemy were blue and red for the blue capability network and red and blue for

the red capability network. Since there were cycles that led to no direct capability (e.g., cycles between units communicating information and commands), the measure of cyclicity has to be corrected for these *inert* cycles. The net cyclicity ($\lambda^{\mathcal{A}}_{1,net}$), the cyclicity that was greater than the inert cyclicity, is obtained using Equation 7. By calculating the $\lambda^{\mathcal{A}}_1$ for the joint graph of the four networks and subtracting the $\lambda^{\mathcal{A}}_1$ for the joint communication and control networks (the inert capability), the net cyclicity for each side's functional graph can then be established.

$$\lambda^{\mathcal{A}}_{1,net} = \lambda^{\mathcal{A}}_{1,total} - \lambda^{\mathcal{A}}_{1,inert} \tag{7}$$

Figure 28 is an example of the output obtained from the process previously described. The blue line depicts the net capability cycles, and the red the fraction of red forces killed. Whenever a red unit is eliminated, the blue $\lambda^{\mathcal{A}}_{1,net}$ spikes, indicating that the capability cycle has been closed, but the converse is not necessarily true since the p_{kill} of the blue ASMs and AAMs is less than 100%. Therefore, due to the way the simulation was created, and the functions measured, a blue spike indicates only the possibility of a red casualty. This possibility nonetheless, is a simple multiplier, and therefore, the architecture's ability to satisfy a capability is still possible as long as the p_{kill} can be estimated.

4.5 Time Averaged Analysis of an ABS with the DiMA Technique

This experiment will focus on the ability of the DiMA technique to reproduce the outcome of a more complex ABS by using time averaged EGMs. In order to ensure that the input to the DiMA technique is compatible to the data from the ABS, the time averaged EGMs are created by the ABS. The DiMA technique will then use these EGMs to create engagement matrices, analyze those, and predict which side is victorious, and which units experienced the most casualties. The ABM used will be eDNOsim, with its four types of units, and four main functions, communications (which includes the ability to give orders), sensing (can either be detecting, tracking, or identifying), and engagement (can either be engaged or vectored).

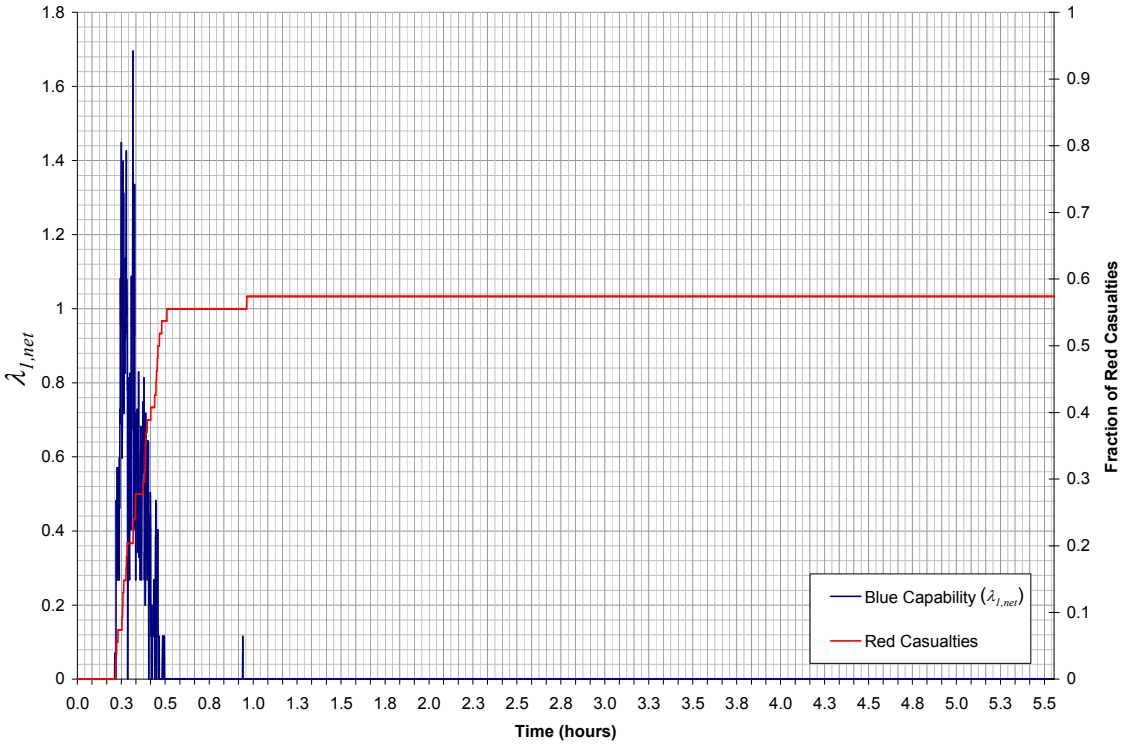

Figure 28: Relation between $\lambda^{\mathcal{A}}_{1,net}$ for the blue force and red casualties.

For the results in this section to support the hypothesis that the DiMA technique is suitable at analyzing large-scale complex architectures, there must be a correlation in one sides casualties and the opposing side's CNE. Specifically, the Pearson's correlation coefficient should be greater than 80% and the p-value less than 0.05.

Figure 29 presents a scatterplot matrix of two eDNOsim outputs (as the rows) and two DiMA outputs as the columns. The correlation between $\lambda^{\mathcal{A}}_{1}$ and enemy casualties is significant. Table 11 contains the Pearson's correlation and p-value for CNE and enemy casualties for both the blue and red sides. The values obtained for the test cases executed with eDNOsim produced correlation coefficients in excess of 80% and with p-values well below the required 5%. This is an encouraging result for the DiMA technique and strongly supports Hypothesis A.

For an additional test, gDNOsim was executed for 10 repetitions in an engagement consisting of 340 agents, of 20 different types, with 195 on the blue side and 145 on the red side. In this case, the functional cyclicity in the enemy's capability cycle as calculated

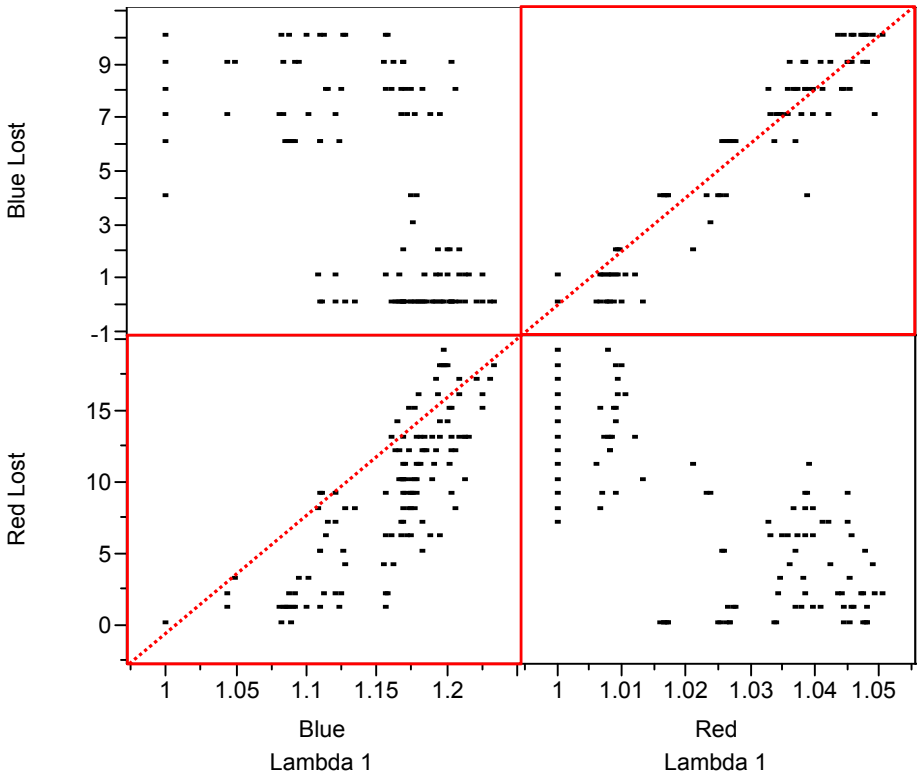

Figure 29: DiMA and eDNOsim results comparison.

Table 11: DiMA and eDNOsim correlations for CNE and units killed.

	Blue CNE — Red Killed	**Red CNE — Blue Killed**
Correlation	84.7%	97.5%
p-Value	4.11×10^{-45}	2.02×10^{-105}

by DiMA for each agent type was tested for correlation against the fraction of casualties for that type. Table 12 contains the Pearson's correlation and p-value for 10 cases. As with the previous test with eDNOsim, the correlation between the functional cyclicity of the agent types as calculated by DiMA using time-averaged data from the gDNOsim is highly correlated to the casualties of that type. That indicates that in relative terms, one could have predicted which types would be more susceptible to higher casualties, and which ones would be more likely to survive the engagement. A direct test of the ability of DiMA to enable relative comparisons can be performed by comparing every pair of casualties as calculated by gDNOsim and every corresponding pair of functional cyclicities calculated by DiMA. If DiMA agrees approximately 50% of the times, then it can be concluded that DiMA cannot predict which entities will suffer more casualties than others. If it agrees considerably less than 50% it means that there is an inverse bias, and it may be indication that functional cyclicity in the enemy's capability cycles as measure by DiMA is inversely correlated with likelihood that an agent type will suffer more casualties. The final outcome, is that the relative results of DiMA and the casualties as measured by the ABS have a high decree of concordance, indicating that DiMA can in fact predict for a most pairs of agent types which ones will suffer the most casualties. The results for 200 data points, that is a total of $19,800$ pairs (the comparisons are invertible and comparisons to itself are not useful, therefore there are $200(100-1)$ possible unique comparisons). Of those, $18,145$ agreed, that equals 98.1% of the times. This result is of great encouragement to DiMA's ability to capture the likelihood of predicting outcomes of complex systems, such as the dynamics displayed by gDNOsim. Furthermore, this result in turns supports Hypothesis A.

4.5.1 The variability of the EGM matrix

Since the ability to use DiMA relies on the ability to capture the characteristics of likely engagements by the EGM, it is important to test the uniqueness of the EGM. Due to the computational expense in the test devised to test the variability in the EGM, a single representative case with 10 repetitions was devised. This test is not meant to be extensive nor conclusive, it is mean to serve as a demonstration and illustration of the concern of

Table 12: DiMA and gDNOsim correlations for functional cyclicity in enemy's capability cycle and fraction of agents dead.

Case	**Correlation**	**Pearson's P-Value**
1	87%	7.69×10^7
2	89%	1.27×10^7
3	88%	2.56×10^7
4	87%	5.41×10^7
5	83%	6.32×10^6
6	89%	1.06×10^7
7	85%	1.57×10^6
8	87%	5.53×10^7
9	85%	1.74×10^6
10	88%	3.40×10^7
All	83%	6.37×10^6

characterization of the model by the time-average EGM formulation. The author recognizes that future work should explore this topic further.

For this test, gDNOsim was used due to its ability to generate a large number of agent types. Each type of agents has different characteristics and these characteristics affect not only the ability of the agents to engage and allow one side to be victorious but also affect the likelihood of functional relations from existing between the types, and therefore affect the EGM.

The eDNOsim and gDNOsim models include a procedure for recording EGMs from the functional connections between the agents in the simulation. At every time step, the existing functional links are queried and the respective entry in the EGM is increased by one for every existing link between any agent of the types for each function. A normalization matrix is also recorded, where for every time-step, for every pair of agent-types, the product of their numbers is added for the respective entry in the matrix. Due to multiplication's commutative property, the normalization matrix is symmetric. This matrix can then be used to normalize the recorded EGM to obtain a time-averaged EGM that can be used by the DiMA technique.

For each repetition, a $20 \times 20 \times 4$ dimensional EGM was measured. The four functions were communications, tracking, vectored, and engagement. The variability of each of the four functions can be studied independently. Figure 30 graphically represents the average probability that any of the four functional interactions occurs between any of the 20 agent types. Each of the four plots represents one of the four functions modeled in gDNOsim. The agent types are listed in the ordinate and the abscissa of each of the four plots, and the colors of the dot are representative of the average probability of the two agents interacting. The interactions are denoted in a clockwise manner, meaning, that the direction for the relation is represented as going from the agents listed on abscissa to the ones on the ordinate. Missing dots indicate that there was no probability of the two agents interacting, e.g., blue and red agents could not communicate with one another. It is important to mention that the agents have been resorted by their side (Blue first, and Red second) for ease of representation, and their values do not map to the ones listed in Table 12.

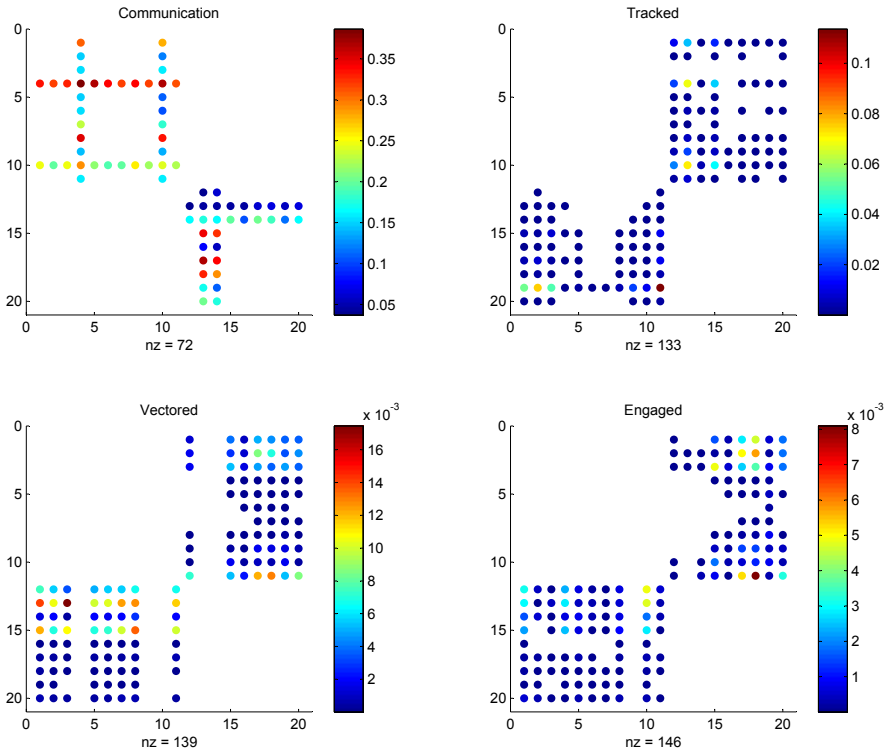

Figure 30: Average probability of functional interaction for the 20 agents types in the gDNOsim simulations.

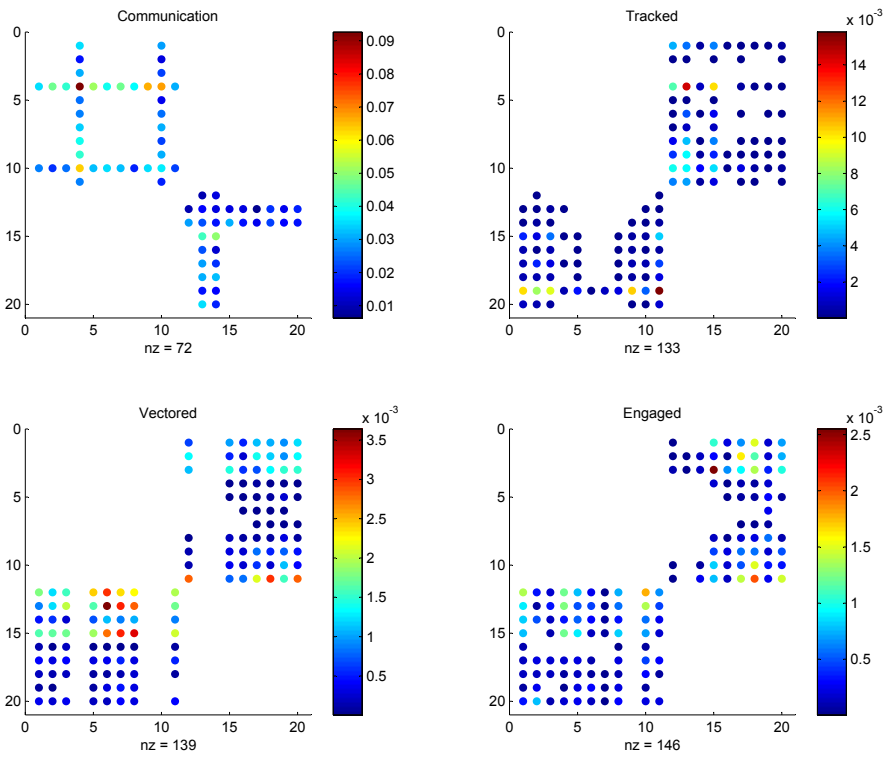

Figure 31: Standard deviation of functional interaction for the 20 agents types in the gDNOsim simulations.

Figure 31 represents the standard deviation for all possible functional interactions, and Figure 32 represents for each possible functional interaction, the ratio between the standard deviation calculated and the average. High values of this ratio, indicate that there is considerable variability when compared to the measured average. Several normalized variabilities exceed unity, indicating that the variability is considerable. In some cases, the variability tends to be focused on some of the units, e.g., for communications, the 13^{th} agent (the 2^{nd} agent in Table 12) seems to display a large amount of relative variability–when compared to the other unit types–in its ability to communicate with other agents. This particular agent type, suffered 100% casualties in 9 out of the 10 repetitions, it suffered an attrition of 87% in the other case. That is the highest attrition rate of all agent types. Furthermore, that agent had the lowest communications range of any other type (8.9). Small variability in the starting location of the agents, would produce large variability in the chances that it would be able to communicate with others. Tracking, Vectoring, and Engaging displayed considerably higher levels of normalized variability in some of their functional relationship pairings.

It may be possible that the variability is due to changes in the agents' survival as the simulation progresses, indicating that the EGMs may be different not necessarily because of their inability to capture behaviors in the model, but due to the chaotic effects of the model itself. For these types of situations, estimating the EGM may prove difficult at best, but the ease by which the DiMA process can be executed, allows users to study large number of variations, including Monte Carlo Simulations of distributions of likelihoods of interactions.

As a final caveat of this section, the author would like to remind the reader that due to the limited resources, only 10 repetitions were simulated, larger numbers may produce more encouraging results, nonetheless, it is concerning that for this simulation, there was such a large variability observed in the construction of the EGM.

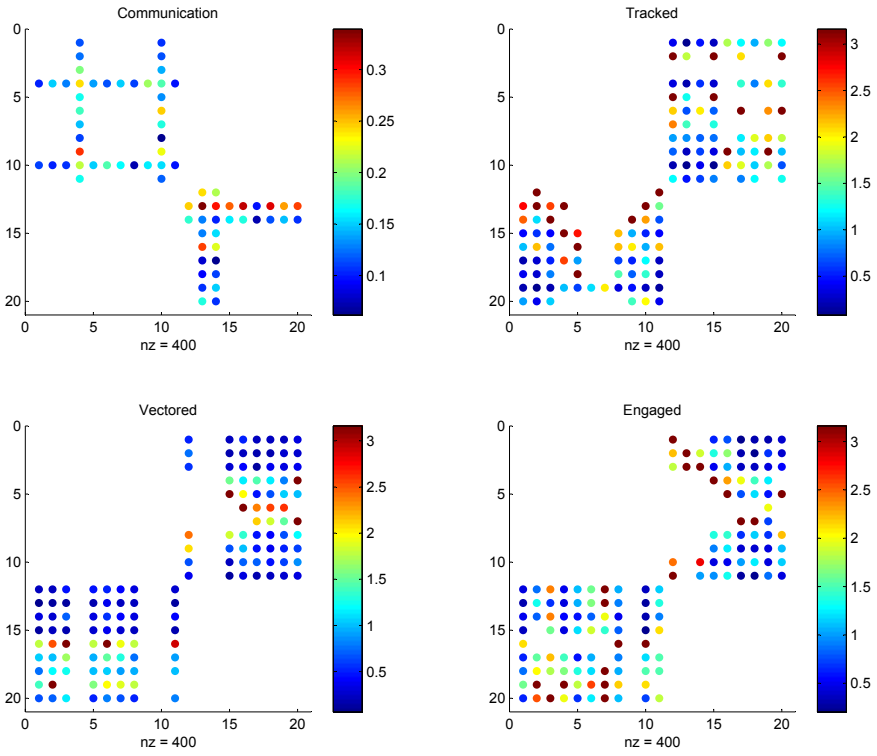

Figure 32: Normalized variability of functional interaction for the 20 agents types in the gDNOsim simulations.

CHAPTER V

IDENTIFYING THE CRITICAL SYSTEMS TO MODEL

"No amount of experimentation can ever prove me right; a single experiment can prove me wrong."

- Albert Einstein

This section will concentrate on testing the hypothesis that a PFE-based ranking of the modeling importance of the entities of an ABM&S provides the best representation of its behavior for a given modeling effort. The intent is not to prove the hypothesis, since as Albert Einstein eloquently stated in the quote presented above, no amount of experimentation can ever prove a hypothesis to be true, yet one single experiment can prove it incorrect. The intent is to create tests that are rigorous and are expected to stress the hypothesis, if the results from the tests agree with the hypothesis, it is not an indication that they hypothesis is true, simply that it has not been proven not true. The only definitive result that can be obtained from testing an inductive hypothesis is a negative one, which would in turn disprove it. All other results simply support the hypothesis.

Measuring how complex the behavior of an agent of an ABM and the overall System-of-Systems (SoS) is not a simple endeavor. There are a large number of degrees of freedom as to how such a thing can be done. In addition, complexity in these applications is more subjective and tends to have less strict definitions than previously discussed. For this reason, using a pure ABM&S formulation to quantitatively test hypotheses of the nature of this one is difficult since the validity of those tests can always be argued since they are based on subjective definitions. A framework that provides analogous behavior, while at the same time allows for strict quantifiable measurement of its properties is therefore essential.

ABM&S in its essence is a formulation based on entities that have a finite set of states, evolve in discrete time steps, interact with a portion of the other agents, and change their

state based on internal rules and the states of the entities they can interact with. By simplifying this description of ABM&S we can identify a framework that provides the required characteristics. If the finite number of states the agents can have is reduced to the non-trivial minimum of two and the interactions between agents is frozen over time, we obtain what is commonly referred to as a Random Boolean Network (RBN).[1] The rules of an RBN are defined by a truth table (or boolean table) that prescribes the next state for any given node based on the state of the nodes that it is attached to. Since this hypothesis is concerned with modeling effort and fidelity, the behavior of two different RBNs most be compared. One RBN will have all the rules "active," this RBN will be referred to as Reference RBN (RBN_R). The second RBN will have a portion of its rules "inactive," this RBN will be referred to as Model RBN (RBN_M). The inactive rules will be set to -1, whereas the active rules will have values of either 1 or 0. The RBN formulation will be modified so when an input state for a node matches that of an inactive rule, the state of the node will not change. This is analogous to not modeling a certain rule in an agent, whereby if an agent faces a condition that is not modeled, it will not change state. Modeling effort is analogous to the fraction of rules that are active in the RBN_M and the total number of rules that are active in RBN_R.

5.1 Node Ranking Schemes

Since the hypothesis stipulates that the functional centrality of the nodes is the best indicator for their impact on the overall's system behavior, other competitive ranking schemes must be tested to ensure that that in fact is the best ranking. Therefore, the nodes will be ranked based on the Perron-Frobenius Eigenvector, Fiedler Vector, in-degree, out-degree, clustering coefficient, core coefficient, uniformly (null hypothesis: ranking is not important), and randomly (null hypothesis: intelligently ranking is not important). The rationale for all these rankings can be considered to be a bottom-up approach, a reductionist approach of sorts, where the different rankings (other than the two null hypotheses) were considered to be competitive candidates by hypothesizing that the characteristics of the nodes they

[1] RBNs were developed by Kauffmann in the 60's to study biochemical systems, for more information, refer to Appendix B.

captured were likely to (1) have an impact on the overall behavior of the networks or (2) be more active because they are more likely to change state, or a combination of the two. All these rankings are inspired from two underlying hypotheses, that nodes that are well connected are more likely to influence or be influenced by other nodes, and nodes that are part of clusters within the network, are more likely to exhibit the faster dynamics and drive the dynamics of the overall network. For a node to be considered important it must not only have fast dynamics (a higher chance of changing state), but to have an impact on the other nodes in the network as well so the complex dynamics can propagate through the network.[2] Balancing these two is not a simple enterprize, and no close form solution exists as to how best to identify these nodes. These rankings will attempt to do this by focusing on different elements of the node characteristics. Below is a more detailed explanation of each of the ranking schemes and the rationale behind them.

PFE The PFE based ranking assigned to each node the value of the eigenvector associated with that node. This is the basis for the PageRank™system that Google™uses to rank websites.[3] The PFE is a measure of centrality of each of the nodes of the graph [57], which means that it not only accounts for how well connected a node is, but also how well its neighbors are connected. It is hypothesized that since nodes that are well connected to other well connected nodes, will have their behavior have a larger impact in the overall behavior of the system. Equation 8 shows how the ranking ($\mathbf{r}$) for each of the n nodes is calculated from the PFE ($\mathbf{x}^{\lambda_1^{\mathcal{A}}}$).

$$\mathbf{r}_i^{p(PFE)} = ||\mathbf{x}_i^{\lambda_1^{\mathcal{A}}}||_1 = \frac{\mathbf{x}_i^{\lambda_1^{\mathcal{A}}}}{\sum_{i=1}^{n} \mathbf{x}_i^{\lambda_1^{\mathcal{A}}}} \tag{8}$$

Fiedler Vector The eigenvector associated with the first non-trivial eigenvalue of the Laplacian matrix is commonly referred to as the Fiedler vector in honor of Miroslav Fiedler, its associated eigenvalue is commonly referred to as the algebraic connectivity

[2] It is well understood that the topology of an RBN dictates the dynamics of their nodes, traditional RBNs have been studied for these properties but their results are only valid for the N-K model.[190, 292, 351]

[3] Google's™search engine ranks websites based on their centrality in a hyperlink graph, where the nodes are the websites, an a directed edge exists between two websites if one has a hyperlink to the other.

of the graph. The logic behind the selection of this ranking is that a node's ability to synchronize (reach a steady-state) is hypothesized to be related to its corresponding value of the FV.[258] Kincaid *et al.* state that for networks with fixed degree distribution and fixed network assortativity[4], "those with small λ_2^{ℓ} are shown to be poor synchronizers, to have much longer shortest paths and to have greater clustering in comparison to those with large λ_2^{ℓ}."[191] Since deviation from steady-state indicates the activity the node will have in the evolution of the network, 'synchronizability' provides a promising ranking. Networks with low values of algebraic connectivity (more highly clustered) and nodes within that network that have values far from zero (both positive and negative), are considered to be less likely to reach steady state and be part of tightly woven clusters. Two rankings based on the FV will be tested, (1) the absolute value of the FV because it will have larger values for nodes with FV values that are away from zero and are more likely to be part of larger clusters and be less likely to reach steady state, and (2) a corrected version of the FV. The absolute FV-based ranking is described in Equation 9, where the Fiedler eigenvector ($\mathbf{x}^{\lambda_2^{\ell}}$) is used to calculate the ranking $\mathbf{r}$ for each of the n nodes.

$$\mathbf{r}_i^{a(FV)} = || \left| \mathbf{x}_i^{\lambda_2^{\ell}} \right| ||_1 = \frac{\left| \mathbf{x}_i^{\lambda_2^{\ell}} \right|}{\sum_{i=1}^{n} \left| \mathbf{x}_i^{\lambda_2^{\ell}} \right|} \tag{9}$$

The secondary ranking based on the corrected FV does not promise to be as suitable as the first, but it will be investigated in case it offers useful results. Since the elements of the FV can range between very large magnitude negative numbers and very large magnitude positive numbers, the value of its elements need to be positive for the 1-norm to produce numbers in the zero to one range. This secondary metric based on the FV, from hereon referred to as the *Corrected Fiedler Vector* ranking, ($\mathbf{r}_i^{c(FV)}$) will be calculated as presented in Equation 10.

$$\mathbf{r}_i^{c(FV)} = ||\mathbf{x}_i^{\lambda_2^{\ell}} - \min_{i=1}^{n} \left(\mathbf{x}_i^{\lambda_2^{\ell}} \right) + \min_{i=\mathbf{x}_i^{\lambda_2^{\ell}}>0} \left(\mathbf{x}_i^{\lambda_2^{\ell}} \right) ||_1 \tag{10}$$

[4]Network Assortativity is a measure of the network's preference to attach to nodes based on a similarity or difference, the measure of similarity or difference most often is the degree of the node.

Clustering Coefficient The clustering coefficient (c^{cluster}) is the ratio of the number of edges between a vertex's neighbors to the total possible number of edges between the vertex's neighbors. The logic behind choosing this ranking is that nodes that have high clustering coefficients probably exhibit the fastest dynamics and should therefore be modeled with the highest fidelity since their states will change the most. Equation 11 describes how the vector of clustering coefficients ($\mathbf{c}^{\text{cluster}}$) is used to compute the ranking $\mathbf{r}$ for the n nodes.

$$\mathbf{r}_i^{p(CC)} = ||\mathbf{c}_i^{\text{cluster}}||_1 = \frac{\mathbf{c}_i^{\text{cluster}}}{\sum_{i=1}^{n} \mathbf{c}_i^{\text{cluster}}} \tag{11}$$

Core Number The core number (n^{core}) of a node i is the largest integer c, such that when all other nodes with degree less than c are removed, node i is still connected. It gives an indication of how well connected a node is, not only to other nodes, but how well the nodes that node is connected are connected to others. As it was the case with the PFE, this "connectedness" of the node may provide a good indication to their critical contribution to the overall dynamics of the RBN. Equation 12 describes how the vector of core numbers ($\mathbf{n}^{\text{core}}$) is used to compute the ranking $\mathbf{r}$ for the n nodes.

$$\mathbf{r}_i^{p(CN)} = ||\mathbf{n}_i^{\text{core}}||_1 = \frac{\mathbf{n}_i^{\text{core}}}{\sum_{i=1}^{n} \mathbf{n}_i^{\text{core}}} \tag{12}$$

In-Degree This ranking assigns to each node the number of edges that reach that node. It is important to remember that the more incoming edges a node has, the more rules it will have, and at the same time, the more likely it is that its state will change since any of its inputs changing is possibly a change of state for that node. The ranking based on the in-degree of each node is presented in Equation 13, where the in-degree ($\mathbf{d}^{\text{in}}$) is used to calculate the ranking $\mathbf{r}$ for each of the n nodes.

$$\mathbf{r}_i^{p(ID)} = ||\mathbf{d}_i^{\text{in}}||_1 = \frac{\mathbf{d}_i^{\text{in}}}{\sum_{i=1}^{n} \mathbf{d}_i^{\text{in}}} \tag{13}$$

Out-Degree This ranking assigns to each node the number of edges that leave that node. The logic behind this ranking is that the more nodes a given node's state affects, the more likely it is that that node will have an effect on the overall behavior of the RBN. Equation 14 describes how the ranking for each of the n nodes is calculated.

$$\mathbf{r}_i^{p(OD)} = ||\mathbf{d}_i^{\text{out}}||_1 = \frac{\mathbf{d}_i^{\text{out}}}{\sum_{i=1}^{n} \mathbf{d}_i^{\text{out}}} \tag{14}$$

Uniform Assigns the same value to all nodes. The purpose of this ranking is to ensure that prioritizing the nodes offers and advantage, by testing the null hypothesis that no prioritization of modeling effort is equally as good as other prioritizations. If this ranking performs as well as any of the others, the experiment will indicate that effort should not be spent in attempting to rank the nodes using the methods described above. The value for each node according to this scheme is given in Equation 15.

$$\mathbf{r}_i^U = \frac{1}{n} \tag{15}$$

Random It randomly assigns a priority to each node. The purpose of this ranking is to ensure that intelligently prioritizing the nodes offers an advantage by testing the null hypothesis. If this ranking does well when compared to any of the others, it means that there is not benefit in expending effort in identifying the most critical nodes, any ranking will suffice. Equation 16 describes how the ranking for each of the n nodes is obtained randomly.[5]

$$\mathbf{r}_i^R = ||\text{random}_{n\times 1}||_1 \tag{16}$$

When these rankings were studied in preliminary studies, the PFE and in-degree rankings performed the worse, producing binary fidelities well below 50%. Since the network is boolean, consistently having a binary fidelity below 50% indicates that the patterns of the Time State Matrix (TSM) are being reversed. Statistically, if the behavior of the RBN_M does not display a bias, it should approach 50% binary error.

[5]The 1-norm must be applied after the random vector is created.

These results led to the consideration that it may be possible that a proportional-based ranking (one in which the importance was proportionally equal to the numerical ranking assigned by the scheme) may not be ideal, but an inversely proportional ranking may be more suitable. For this reason, the PFE, in-degree, out-degree, the clustering, and core rankings were inverted. Furthermore, by inverting these rankings, a comparison between the performance of the proportional and inverse rankings can serve as an indicator of how well the ranking captures the criticality of the entities. If the ranking that produces the best results produces the worst results when inverted, this is stronger support for the ranking. Inversion of the rankings consisted in subtracting their value from the maximum and adding the minimum as described in Equation 17. In this case $\mathbf{x}$ is a placeholder for any vector used to create the ranking for the nodes.

$$\mathbf{r}_i^{i(\mathbf{x})} = \max_{i=1}^{n}(\mathbf{x}_i) - \mathbf{x}_i + \min_{i=1}^{n}(\mathbf{x}_i) \tag{17}$$

$$\mathbf{r}_i^{c(PFE)} = ||\mathbf{x}_i^{\lambda_1^A} + \text{median}(\mathbf{x}_i^{\lambda_1^A})||_1 \tag{18}$$

A specific case was added for the PFE based ranking. Since the possibility existed that the zero ranking of nodes may have a detrimental effect in how the rules are discarded,[6] an additional ranking was added. This ranking was computed by adding the median of the PFE to the PFE and taking the 1-norm, from hereon it will be referred to as the *Corrected PFE* ranking. Equation 18 describes how this was done. Table 13 contains the 15 rankings chosen for the experiments, along with their acronym and equation.

All the rankings were normalized by the 1-norm to ensure that the value for their sum over all the nodes would remain constant and the proportionality of rules between the alternatives would remain constant as well. Figure 33 represents the characteristic shapes of the original node ranking schemes used. The nodes (along the abscissa) are sorted for each ranking from lowest ranking to highest ranking. Since the nodes were sorted differently

[6]By having nodes with rankings of zero, it is possible for the rule simplification scheme to disregard all their rules, even in the rule proportional simplification case. This may produce RBNs that behave considerably worse to RBNs from rankings that assigned at least a minimum value to all their nodes.

Table 13: Node Ranking Schemes Evaluated.

Name	**Symbol**	**Equation**
Proportional PFE	$\mathbf{r}^{p(PFE)}$	$\|\|\mathbf{x}_i^{\lambda_1^A}\|\|_1$
Inversely Proportional PFE	$\mathbf{r}^{i(PFE)}$	$\|\|\max_{i=1}^{n}\left(\mathbf{x}_i^{\lambda_1^A}\right)-\mathbf{x}_i^{\lambda_1^A}+\min_{i=1}^{n}\left(\mathbf{x}_i^{\lambda_1^A}\right)\|\|_1$
Corrected PFE	$\mathbf{r}^{c(PFE)}$	$\|\|\mathbf{x}_i^{\lambda_1^A}+\text{median}(\mathbf{x}^{\lambda_1^A})\|\|_1$
Corrected Fiedler Vector	$\mathbf{r}^{c(FV)}$	$\|\|\mathbf{x}_i^{\lambda_2^L}-\min_{i=1}^{n}\left(\mathbf{x}_i^{\lambda_2^L}\right)+\min_{i=\mathbf{x}_i^{\lambda_2^L}>0}\left(\mathbf{x}_i^{\lambda_2^L}\right)\|\|_1$
Absolute Fiedler Vector	$\mathbf{r}^{a(FV)}$	$\|\|\left\|\mathbf{x}_i^{\lambda_2^L}\right\|\|\|_1$
Proportional Clustering Coeff.	$\mathbf{r}^{p(CC)}$	$\|\|\mathbf{c}_i^{\text{cluster}}\|\|_1$
Inversely Prop. Cluster. Coeff.	$\mathbf{r}^{i(CC)}$	$\|\|\max_{i=1}^{n}\left(\mathbf{c}_i^{\text{cluster}}\right)-\mathbf{c}_i^{\text{cluster}}+\min_{i=1}^{n}\left(\mathbf{c}_i^{\text{cluster}}\right)\|\|_1$
Proportional Core Number	$\mathbf{r}^{p(CN)}$	$\|\|\mathbf{n}_i^{\text{core}}\|\|_1$
Inversely Prop. Core Number	$\mathbf{r}^{i(CN)}$	$\|\|\max_{i=1}^{n}\left(\mathbf{n}_i^{\text{core}}\right)-\mathbf{n}_i^{\text{core}}+\min_{i=1}^{n}\left(\mathbf{n}_i^{\text{core}}\right)\|\|_1$
Proportional In-Degree	$\mathbf{r}^{p(ID)}$	$\|\|\mathbf{d}_i^{\text{in}}\|\|_1$
Inversely Prop. In-Degree	$\mathbf{r}^{i(ID)}$	$\|\|\max_{i=1}^{n}\left(\mathbf{d}_i^{\text{in}}\right)-\mathbf{d}_i^{\text{in}}+\min_{i=1}^{n}\left(\mathbf{d}_i^{\text{in}}\right)\|\|_1$
Proportional Out-Degree	$\mathbf{r}^{p(OD)}$	$\|\|\mathbf{d}_i^{\text{out}}\|\|_1$
Inversely Prop. Out-Degree	$\mathbf{r}^{i(OD)}$	$\|\|\max_{i=1}^{n}\left(\mathbf{d}_i^{\text{out}}\right)-\mathbf{d}_i^{\text{out}}+\min_{i=1}^{n}\left(\mathbf{d}_i^{\text{out}}\right)\|\|_1$
Uniform	$\mathbf{r}^{U}$	$\frac{1}{n}$
Random	$\mathbf{r}^{R}$	$\|\|\text{random}_{n\times 1}\|\|_1$

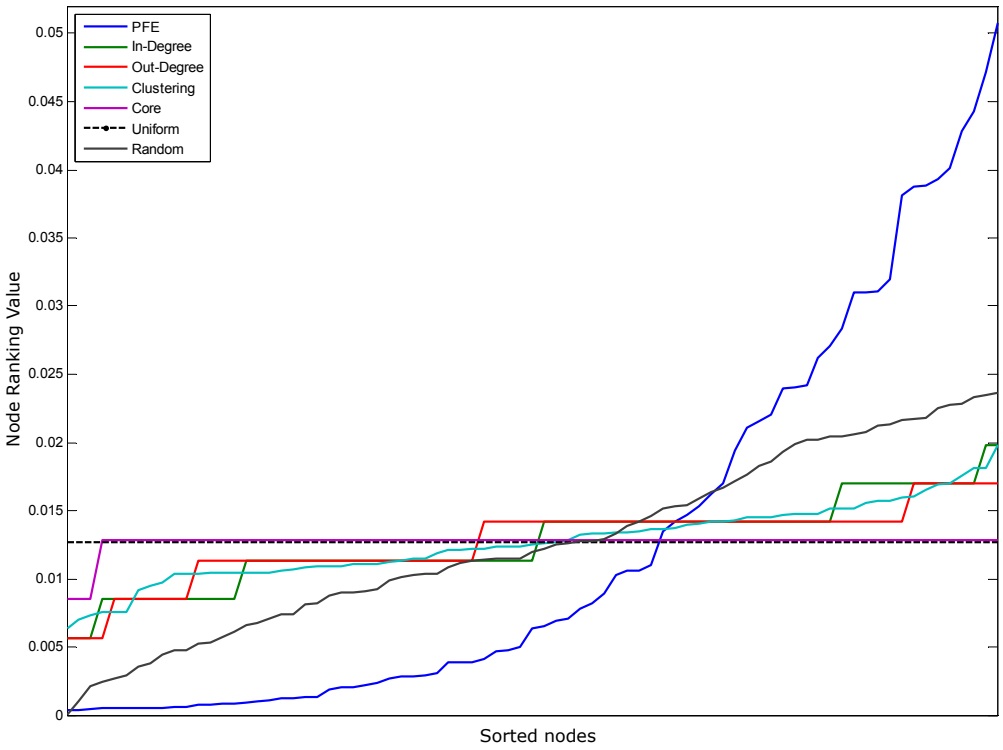

Figure 33: Example of node rankings according to the proportional schemes.

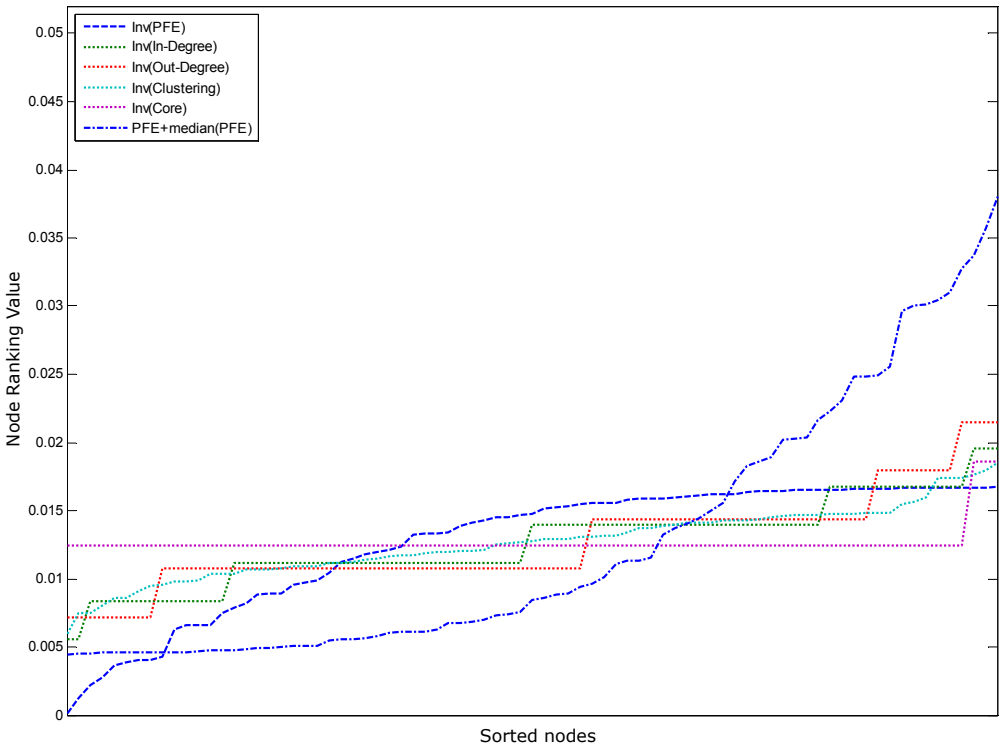

Figure 34: Example of node rankings according to the adapted schemes.

for each ranking, the values on the abscissa do not refer to the same nodes for all rankings. These charts are meant to illustrate the distribution of value by the different rankings. Some rankings distribute the prioritization in an egalitarian fashion, e.g., $\mathbf{r}^{p(CN)}$ assigned almost the same value to all the nodes, whereas $\mathbf{r}^{p(PFE)}$ had the greatest skewness, assigning most of the value to a few of the nodes. This means that for the proportional PFE-based ranking, a large number of nodes have a ranking of near zero. This means that some nodes will have the majority of their rules ignored. The question is then: does having nodes with a ranking of zero reduce fidelity? Figure 34 depicts an example of the node rankings for the adapted schemes (the inverse one plus the median PFE). The inverse of the in-degree, out-degree, and clustering rankings and the median PFE seem almost identical to the original ones. This indicates that the effect they may have is dependent on the selection of the nodes and not the shape of the distribution of prioritization. The inverse of the PFE and core rankings are distinctly different from their originals so not only the selection of nodes may affect their ability to select which rules to disregard, but the way the prioritization is distributed may contribute to the selection of the rules.

5.2 Metrics

A series of metrics will be studied for each RBN to (1) characterize the network, and (2) compare the goodness of the different node rankings. The metrics are categorized into four groups, structural metrics (to study the relationship between structural and dynamic characteristics), rules metrics (to study the impact that the rules have on the behavior of the RBN, complexity metrics to identify correlations between the previous characteristics and the complexity of the behavior produced by the RBN, and the fidelity of the Model RBN (how well the behavior of the Reference RBN (RBN_R) is matched by each ranking and their associated model RBNs (RBN_M)).

5.2.1 Structural Metrics

Structural metrics are the metrics related to the topology of the RBN. A wide variety of metrics related to the structure of the graph will be studied. The structural complexity of the graph will be characterized by the *Off-Diagonal Complexity* of the adjacency matrix

[85] which provides more meaningful valuations of the complexity than entropy and degree distribution measures. More information is provided in appendix D.4. The cyclicity of the graph will be measured using the largest eigenvalue of the adjacency matrix (λ_1), and the Fiedler eigenvalue of the Laplacian matrix (also known as algebraic connectivity) of both the directed ($\lambda_{2,d}$) and undirected ($\lambda_{2,u}$) graph will provide insight into the ability of the network to synchronize its behavior. The number of components (connected partitions of the graph) and edges will also be tracked.

5.2.2 Rules Metrics

The number of rules in the RBN_R will be tracked, as well as the entropy of its rule matrix (a matrix that contains all the rules of the RBN, padded with zeros to make the matrix rectangular). The entropy ranges between zero and 1, while the number of rules is an integer greater than zero. The fraction of active rules in RBN_M will also be tracked, as well as the entropy of its rule matrix. In this case, the matrix will be padded with

5.2.3 Complexity Metrics

The complexity of the overall RBN can be computed algorithmically by determining the Algorithmic Information Complexity (AIC), also known as the Kolmogorov complexity, or Kolmogorov-Chaitin Complexity. This measure of complexity determines the length of a computational element capable of representing the output of the system of interest, in this case the TSM of the RBN. The more ordered the output, the simpler the program, the more disordered, the longer the program, with the limit that a completely random sequence will need at least its length to reproduce it. It has been shown [307, 229, 80] that the general algorithm to estimate the true Kolmogorov complexity of a finite sequence cannot be given because there are an infinite number of ways in which such a program could be constructed and there is always the possibility that a simpler algorithm exists. Lempel and Ziv [209] developed an algorithm to estimate the Kolmogorov complexity of a finite sequence by using only two operations, copy and insert, which was then adapted by Kaspar and Schuster.[186] The problem lies in that their algorithm only allows for a single string of zeros and ones, meanwhile, the TSM has one binary string for each node. Nevertheless, it was observed

through experimentation, that if the TSM of the RBN was recoded into decimal form (with values ranging between zero and one) the complexity of the string could be estimated using the Kaspar and Schuster algorithm.

As Lempel and Ziv pointed out, the definition of complexity depends on the lexicon used, and it is therefore not a truly strict measure if no language is established.[309] An alternative to Lempel and Ziv's approach is to use a compression algorithm that searches for patterns in data and replaces them with a simpler term. The analogy to the computational element is the dictionary stored by the compression algorithm that best describes the input string. This method is not perfect in that the compression algorithm may not recognize a pattern, but the efficiency of modern compression algorithms ensures that the possibility of this is minimized to acceptable levels. Therefore, in addition to the Lempel-Ziv algorithm, a compression algorithm based on the `GNU ZIP`, or simply `GZIP`, compression tool, which is freely available with `JAVA` under the `UTIL.ZIP` class will be utilized to compute the AIC. A final approach consists in identifying the information entropy of the TSM.[7]

5.2.4 Fidelity Metrics

Fidelity is measured as how much the behavior of RBN_R is matched by RBN_M. As described previously in the background section, modeling can be taken from two different perspective, reductionist modeling and holistic modeling. The two perspectives imply different meanings of fidelity. Reductionist fidelity focuses in emulating reality as closely as possible, in this case it would mean that the output of the RBN_R should be matched as closely as possible by the output of the RBN_M. Holistic fidelity is a more abstract concept, it means that certain behaviors of RBN_R are represented in RBN_M. It means that traits of the full model are captured by the simpler model. It is more difficult to create algorithms to identify these traits than to compare exact behaviors, furthermore, determining what constitutes a trait (especially in the case of an RBN) is not a trivial pursuit.

[7]For more information on quantifying the complexity of the TSM, please refer to the appendix D.3.1.

5.2.4.1 Reductionist Fidelity Metrics

Reductionist fidelity is analogous with what will be referred to from here on as *Binary Fidelity*. Since the purpose of the reductionist fidelity metrics is to measure how well the model emulates the reference RBN, the unmodified output of the RBN should be used as the basis for this metric. Binary fidelity is calculated by summing for each node all the instances in which RBN_M matched RBN_R and dividing it by the product of the number of nodes, n, and the number of time steps the network was executed, t_{max}. This means that if RBN_M matched RBN_R perfectly, its binary fidelity will have a value of 100%. If it completely misrepresented RBN_R, then its binary fidelity will have a value of 0%. The initial state ($t = 0$) is neglected in all the fidelity computations, meaning that the comparisons are only done after the RBN has been evolved at least once.

After careful study of this metric, an important observation was made. As the networks are evolved, their states tend to diverge due to small errors. If the RBN_M's fidelity was to be compared at 3000 vs 100 evolutions, it is clear that the 100 evolutions of the network would be more accurate. If the time scale of evolution is made too large, comparisons between different RBN_Ms becomes a trivial matter of chance, too small and truly good RBN_Ms cannot distinguish themselves from mediocre ones. For these reason, the matching of binary states is exponentially weighted over time. The weighting applied to each time step is equal to $e^{(t-1)/r}$ where t is the evolution time step, starting with 1, and r is a regularization parameter set to 30. Equation 19 describes how *Binary Fidelity* is computed.

$$\text{Fidelity}_{\text{Binary}} = \frac{1}{n \sum_{t=1}^{t_{max}} e^{\frac{t-1}{r}}} \sum_{\text{Node}=1}^{n} \sum_{t=1}^{t_{max}} e^{\frac{t-1}{r}} \left(\text{TSM}_{\text{R}}(n,t) == \text{TSM}_{\text{M}}(n,t)\right) \tag{19}$$

5.2.4.2 Holistic Fidelity Metrics

Holistic models are not meant to be predictive, but to help understand complex causalities and more abstract patterns. Holistic models are more useful for drawing analogies from than for predicting behavior. To quantitatively measure the capability to draw analogies is not a trivial problem. Nonetheless there is an analogous case that can be drawn for RBNs.

Matching unique mappings of the binary state of the RBN are the focus of the reductionist fidelity metrics. Holistic fidelity metrics will focus in reproducing more abstract behaviors, such as node activity, node correlation, and frequency characteristics of the RBN.

Problems of identification can have two types of error, Type I (or False Positive) error and Type II (or False Negative) error. False Positive is the identification of something when in reality it is not there. False negative is not identifying something that is there. These errors can be made a percentage by dividing by the total number of possible Type I and Type II errors, so as to have a ratio ranging between zero and one. This is often referred to as False Positive Rate (FPR) and False Negative Rate (FNR). FPR is normalized by the total number of observations that were not identified to be positive by the real system and FNR is normalized by the total number of observations that were identified to be positive by the real system. This works well when the normalizing parameter is roughly the same order of magnitude as the false positive or false negative instances. When this is not the case, the rate can often become trivially small, for this reason, normalization of some of the metrics will be done in a different manner.

Node Activity Node activity is measured as how many times a node changed state throughout the evolution of the RBN. Superactive nodes considered to be those that changed state more than the median. The two errors measured are FPR and FNR. These two metrics are calculated in the MATLAB function `RBN.m` described in appendix F.2.1.

Node Correlation Node correlation is defined as the nodes in the network whose binary behavior is deemed to be correlated. The metric used to identify correlation is Pearson's linear correlation coefficient, where a p-value lower than 0.05 is deemed to indicate a correlation sufficiently greater than zero. The two errors measured are FPR and FNR. These two metrics are calculated in the MATLAB function `getNodeCorrelation.m` described in appendix F.2.16.

Spectrum Analysis The 2-Dimensional binary TSM can be encoded in different ways to transform it into a 1-Dimensional vector. This encoding can be unique (or reversible),

or non-unique (irreversible). This means that every binary state of the RBN can be represented as a number, if the encoding is reversible, it means that that number uniquely defines that binary state, and therefore the binary state can be obtained from that single number, can be reversed from it. Non-unique encodings do not have this property, multiple binary states can lead to a single encoding and therefore it is not possible to reverse the binary state from the encoded number. The frequency related metrics described below rely on these 1-D encodings of the TSM to obtain a time-state vector which can then be studied to obtain insight into the frequencies that dominate the behavior of the RBN.

Three metrics are tracked in the frequency spectrums of the encodings as obtained by computing their Fast Fourier Transform (FFT). Two of these metrics are related to the identification of the peak frequencies of the RBN. Peaks in the frequency spectrum describe behavioral characteristics of the RBN which are related to its periodicity and ability to synchronize. Since this is an identification problem, there can be two types of errors, but in this case, error rates are not useful since the normalizing parameters are orders of magnitude larger than the magnitude of false positive and false negative instances. For this reason the error is computed as the difference in the amplitude of the peaks divided by the amplitude of the corresponding peak. False positive error is calculated as the sum of the absolute differences of the amplitude of the peaks identified by RBN_M divided by their amplitude and the number of peaks identified. False negative error is calculated as the sum of the absolute differences of the amplitude of the peaks identified by RBN_R divided by their amplitude and the number of peaks identified. The third metric is concerned with matching the spectrum and is calculated as one minus the integral of the difference between the RBN_R and RBN_M spectrums, divided by the total area under the spectrum of RBN_R. Due to the discrete nature of the spectrum, a trapezoidal integration is sufficient. The MATLAB function used to calculate these three metrics is presented in appendix F.2.17.

1. **Decimal Encoding** The decimal encoding of the binary TSM is unique and

reversible (assuming that sufficient accuracy is allocated to the number of bits (nodes) the RBN has). To encode the binary TSM into decimal form, the nodes are sorted according to the sum of their state, then the TSM is transformed from gray coding to regular binary,[8] and finally this binary encoding is translated to a single decimal number ranging from zero to one. The metrics based on the spectrum of this encoding are obtained as described above.

2. **Hamming Distance** Hamming distance can be used to study the behavior of RBNs as demonstrated by Castro e Silva et al. [76]. Hamming distance for each time step is calculated as the fraction of the number of nodes that have a state different from their initial one. This is a non-unique, irreversible, encoding because the individual behavior of the nodes is not tracked. The metrics based on the spectrum of this encoding are obtained as described above.

3. **Power** Wolfram [345, 347] proposed the use of the "power" of a cellular automaton to study its behavioral characteristics, the same can be done for an RBN. Power in this case is computed as the fraction of nodes that are in an active state. This is also a non-unique, irreversible, encoding of the TSM. The metrics based on the spectrum of this encoding are obtained as described above.

5.2.5 Regret Analysis

The concept of Regret Analysis [266] provides a useful framework for comparing the different structural-based alternatives for node ranking. In order to support the hypothesis that the PFE-based ranking provides the most insight into the behavior of the complete system for a given amount of modeling effort, the results of these tests should show that the PFE-based ranking has the minimum regret under a variety of conditions. Regret Analysis is only as good as the alternatives that are being considered. Therefore, if the alternative rankings are not competitive, the analysis and results are meaningless. The meticulous selection

[8] The reason for sorting the nodes and then decoding the binary strings using gray coding, is to avoid noisiness in the decimal interpretation. The sorting groups the nodes according to their average state and the gray coding ensures that small binary changes, translate to small decimal changes. More information about this process is provided in appendix C.1.

of alternative rankings was therefore a critical step in the process and one that should be considered when using these results.

Equations 21 and 21 describe how regret was calculated for a given metric (m) for an alternative i out of n alternatives. The reason regret for minimizing objective functions can be computed using Equation 21 is that the metric m has a range between zero and one.

$$\text{Maximization: } Regret(m)_i = \frac{\max_{i=1}^{n}(m_i) - m_i}{\max_{i=1}^{n}(m_i)} \tag{20}$$

$$\text{Minimization: } Regret(m)_i = \frac{m_i - \min_{i=1}^{n}(m_i)}{1 - \min_{i=1}^{n}(m_i)} \tag{21}$$

In order for regret analysis to provide meaningful results, the values of the alternatives compared are only evaluated under the same conditions. That means that for every RBN executed, the regret of any alternative can only be computed from the block of metrics obtained from that same RBN. That means that if 15 alternatives are studied for 600 different RBNs, there will be 600 regret "blocks," where each block will have 15 values for each metric of interest. For more details into how regret is calculated, please refer to the MATLAB function `getRegret.m` described in the appendix F.2.19.

The fidelity metrics can be studied and aggregated using Regret Analysis. Two average regret metrics will be pursued in this analysis, one for the reductionist modeling group, and another for the holistic modeling group. These metrics will be referred to as Reductionist Modeling Regret (RMR) and Holistic Modeling Regret (HMR) respectively.

5.3 Results and Analysis

The goal of these experiments is to test that a PFE-based ranking of the nodes will provide the highest fidelity and matching of complexity for a wide variety of RBNs. The topology of the RBNs will be varied in three ways: number of nodes (30-100), network type (Erdős-Rényi G(n,p) model, Watts-Strogatz Small-World Model, and a truncated[9] Barabási-Albert

[9]The reason the model must be truncated is that nodes cannot have a large number of inputs since the length of the rule required to specify their state is equal to $2^{\#\text{ of inputs}}$, the maximum number of inputs to a node is set to 10, which equates to a maximum rule length of 1024 bits.

Scale-Free Model), and the sparsity of the adjacency matrix (4%-8%). The network will be a directed one meaning that the adjacency matrix will not be symmetric in order to increase generality of the results. Two types of rule generation will be pursued, single-rule (where all the nodes have a portion of the same rule and the node(s) with the most inputs have the full rule), and a multi-rule approach, where every node has a different random rule. The final parameter that will be controlled will be the fraction of rules to be discarded (15%-50%).

The power of an experiment like this over a demonstration using one or a handful of ABM&Ss is that a large number of different RBNs can be studied and that the results obtained can have statistical significance. This avoids suspicion that the single (or small number of demonstrations) used to support the hypothesis were tailored in any way, and provides broader means for invalidating the hypothesis. To ensure that the alternative space is studied as thoroughly as possible, a custom DOE will be used which will consist of a 49 case CCD and a 551 case Latin hypercube, producing a total of 600 cases. The Latin hypercube was created in MATLAB using 200 iterations of the `lhsdesign.m` function, with an objective for minimum correlation. Due to the discrete nature of some variables (i.e., type of network, and type of rules), the continuous output of the Latin hypercube had to be rounded off to the nearest discrete value for those two metrics. This increased the correlation between the controlled parameters, but not sufficiently to exceed a threshold of concern. The maximum correlation is 6.89% with a mean correlation between parameters of 1.96%, the minimum p-value is 9.18%, which is above the 5% threshold used to ensure a correlation significantly different from zero.

At the same time, the RBNs' initial state is determined randomly. Therefore, to obtain statistical significance into the ability of a given RBN_M to model the behavior of an RBN_R, both should be initialized from the same initial conditions a number of times. This led to the need to run repetitions for each case in the DOE. Five repetitions were deemed sufficient to identify the means, although more would be ideal.

5.3.1 Test I: Reductionist Modeling Regret

If hypothesis H1.1[10] is correct, this test would show that regardless of the complexity of the system being modeled, an inverse PFE ranking of the nodes would have the least regret of any of the alternatives chosen. Figure 35 depicts the mean RMR for each node ranking scheme under three groupings of regret for four different types of RBN_R's complexity. The blue and red bars are for Rule Proportionality and Node Proportionality respectively. In the areas of low regret (colored in green), high bars mean that that scheme was successful. Yellow depicts the areas of medium regret, and red the areas of high regret. For a scheme to be truly successful, it would have high bars in green and low in all the others. The highest ranking schemes in order are the *Inverse PFE*, *Inverse In-Degree*, *Absolute Fiedler Vector*, and *Inverse Core Number*. The *Inverse In-Degree* approaches the goodness of the *Inverse PFE* to the point where their difference is not statistically significant. The *Absolute Fiedler Vector* ranking at times seems almost as good as these two, but as the system become more complex (Highly Complex and Chaotic), its regret increases, while the first two do not. Finally, the *Inverse Core Number* scheme behaves in a similar fashion to the *Absolute Fiedler Vector* ranking but with slightly lower goodness. The fact that the PFE and In-Degree based rankings are consistently the best and maintain a lead when the system complexity increases, means that if the purpose is to *predict* the behavior of the system the modeling effort should be invested using the inverse PFE ranking scheme.

5.3.2 Test II: Reductionist Modeling Fidelity

Since regret analysis is only useful for comparing alternatives, it is important to analyze the behavior of the different ranking schemes in absolute terms. Figure 36 depicts an image similar to Figure 35, where Binary Fidelity replaced RMR. The reader is reminded that in this case, a high value is desired and therefore the color coding of the rows has been reversed. In this case, the effect that the complexity of the reference model has on the simplified model's ability to represent its behavior is made clear. For very chaotic systems,

[10] A PFE-based ranking of the nodes provides the most suitable guideline for concentrating the modeling effort in order to obtain insight in the behavior of a system.

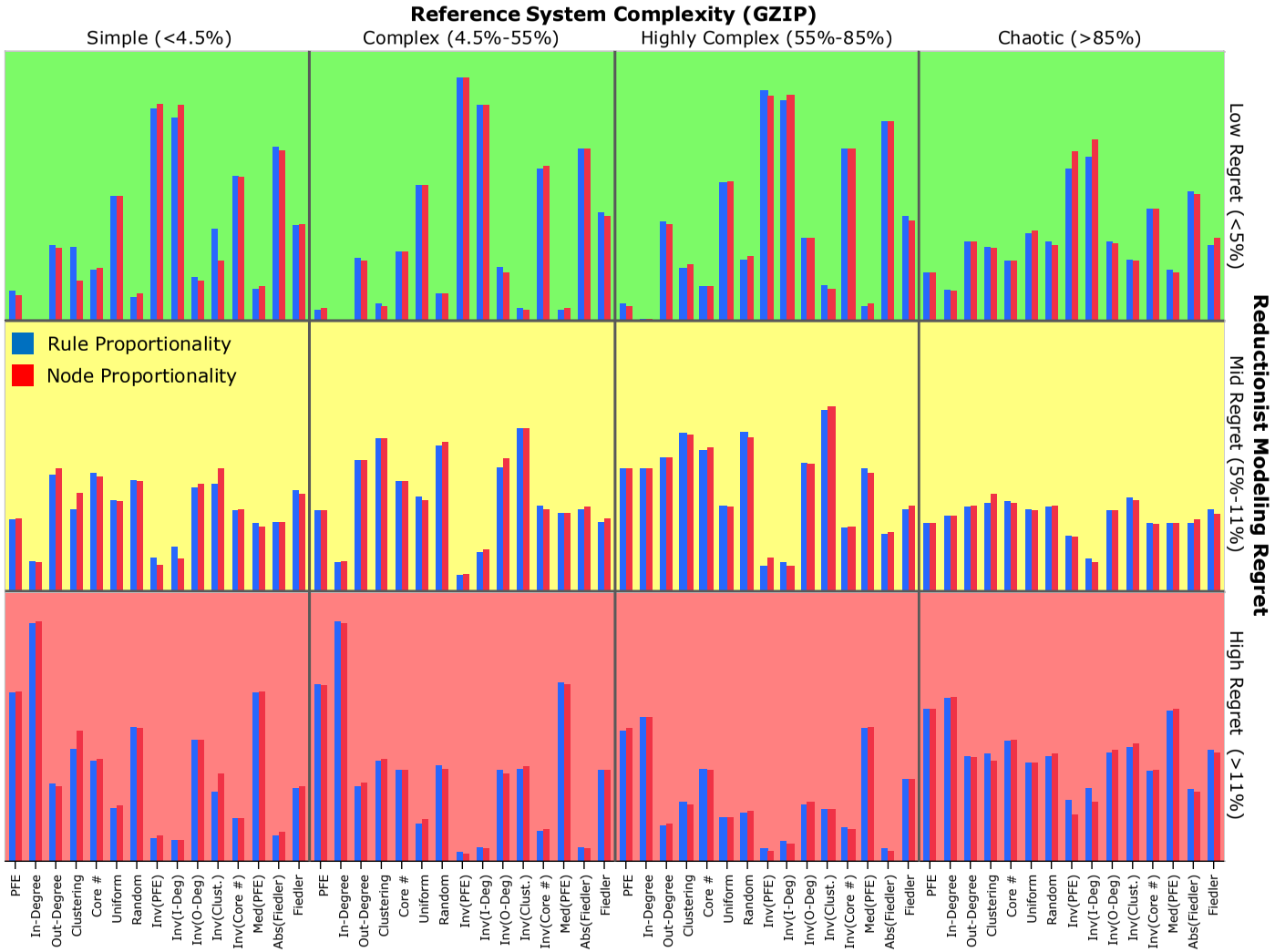

Figure 35: Reductionist Modeling Regret for different node ranking schemes vs. Reference Random Boolean Network complexity.

most rankings do badly, meaning they have low binary fidelity. *Inverse PFE* and *Inverse In-Degree* are the distinctly different ones, with lower number of cases in the low fidelity region, a sizable number of the cases in the medium fidelity, and the only two with at least some cases in the high fidelity. For simple systems, most rankings do fairly well, with *Inverse PFE*, *Inverse In-Degree*, and *Absolute Fiedler Vector* being the only ones that have most of the cases in the high fidelity region. For the complex systems (second column from the left), *Inverse PFE* produces the best results, followed by *Inverse In-Degree*, *Absolute Fiedler Vector*, *Inverse Core Number*, and surprisingly, *Uniform*. Despite the fact that *Uniform* ranking performed surprisingly well, *Inverse PFE* was still superior by a considerable margin. For Highly Complex systems, once again *Inverse PFE* and *Inverse In-Degree* outperformed all rankings, followed by the *Absolute Fiedler Vector* ranking. The portion of cases that have the high fidelity is greatly reduced, but this was expected, as the systems become more complex, predicting their behavior exactly becomes less likely.

5.3.3 Test III: Reductionist Modeling Regret and Effort

Modeling effort has previously been defined as the fraction of rules that are modeled. Figure 37 presents a similar chart as Figure 35 with the different that instead of comparing the regret of the different ranking schemes under different levels of system complexity, the rankings are compared to different levels of modeling effort. The intriguing trend presented in this chart is that as modeling effort decreases (fewer rules are modeled), the *Inverse PFE* and *Inverse In-Degree* seem to do better, but the reader is reminded that this is just the regret, which is a relativistic result. What this chart indicates is that as the modeling effort decreases, *Inverse PFE* and *Inverse In-Degree* outperform the alternative rankings by a wider margin. Looking at binary fidelity and modeling effort, this is clearly illustrated. Figure 38 depicts this relationship for the different node rankings. Note that as modeling effort decreases, binary fidelity decreases as well for all the rankings. The degradation is less for *Inverse PFE* and *Inverse In-Degree* than for the alternative rankings, but there is nonetheless, degradation in the binary fidelity.

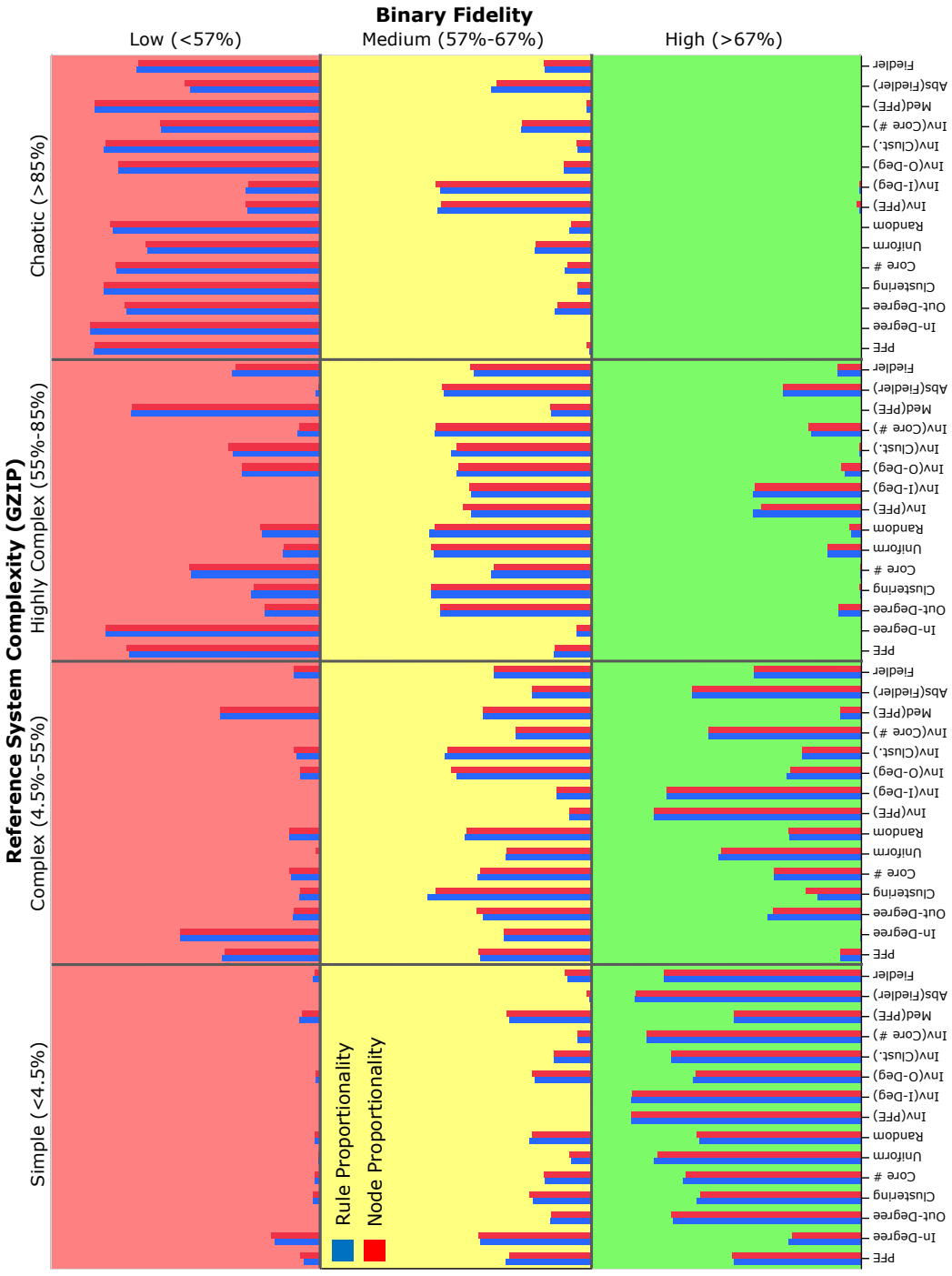

Figure 36: Binary Fidelity for different node ranking schemes vs. Reference Random Boolean Network complexity.

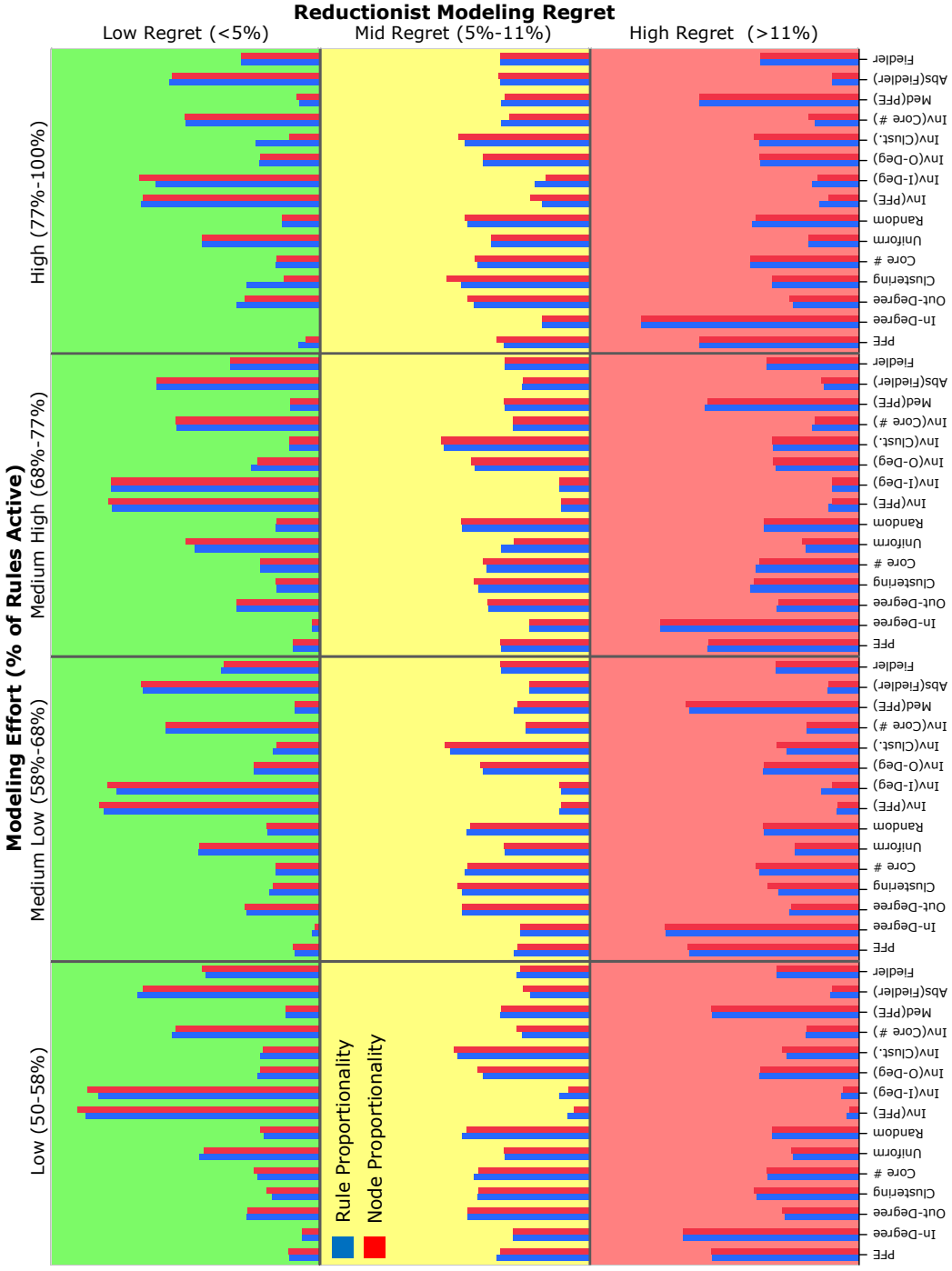

Figure 37: Reductionist Modeling Regret for different node ranking schemes vs. modeling effort.

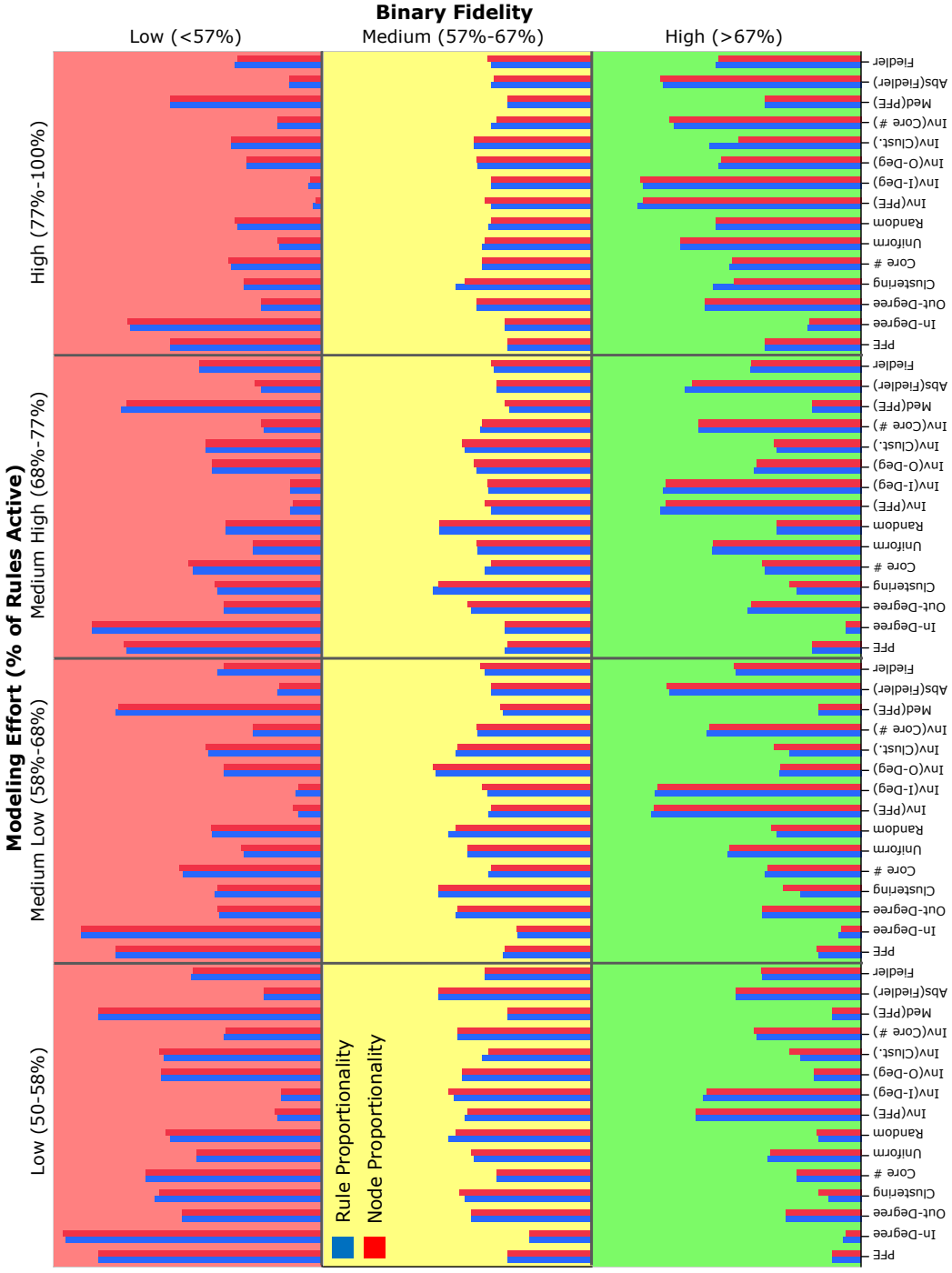

Figure 38: Binary fidelity for different node ranking schemes vs. modeling effort.

5.3.4 Test IV: Holistic Modeling Regret

Holistic modeling is not concerned with predicting behavior, or matching a pattern exactly, but in providing insight into the behavior of the system, or describing patterns in a more abstract manner than simple direct comparison. HMR is a function of the errors (Type I and Type II errors) of the decimal encoding of the TSM, and its the Hamming distance, and power, as well the identification of superactive nodes, and node correlations. There are a total of 5 general characteristics measured, each of which has the two types of error. This metric is therefore an average of a larger number of values, and is therefore more likely to produce results that are not as distinct as the RMR metric.[11] Figure 39 presents the regret that the different node ranking alternatives have for various levels of system complexity. As is the case with RMR, holistic modeling indicates that the *Inverse PFE* and *Inverse In-Degree* produce robust selection of the critical nodes over a wide range of system complexities for both rule proportionality and node proportionality. This indicates that this ranking is suitable for both predictive and understanding purposes, meaning that one simplified model can minimize both holistic as well as RMR. As mentioned previously, these results are only valid when compared amongst the 15 rankings used in these tests. The hypothesis that a PFE-based ranking produces the simplified RBN with the highest fidelity is supported.

5.3.5 Test V: Correlation in the ranking schemes

After studying the ability of the different ranking schemes in identifying the critical nodes to be modeled, the question of whether they are equally capable by ranking nodes differently, or they are equally capable in achieving the same rankings? Since the *Inverse PFE* and *Inverse In-Degree* rankings seem to behave orders of magnitude better than the alternatives, is this indication that they are selecting the same nodes?

To study this and the general correlation problem for the different ranking schemes, a one hundred random network experiment was devised, where the ranking of the nodes

[11] By being an average of a larger number of values, this metric has the potential to have less variability since not a single node ranking may be clearly dominant along every sub-metric.

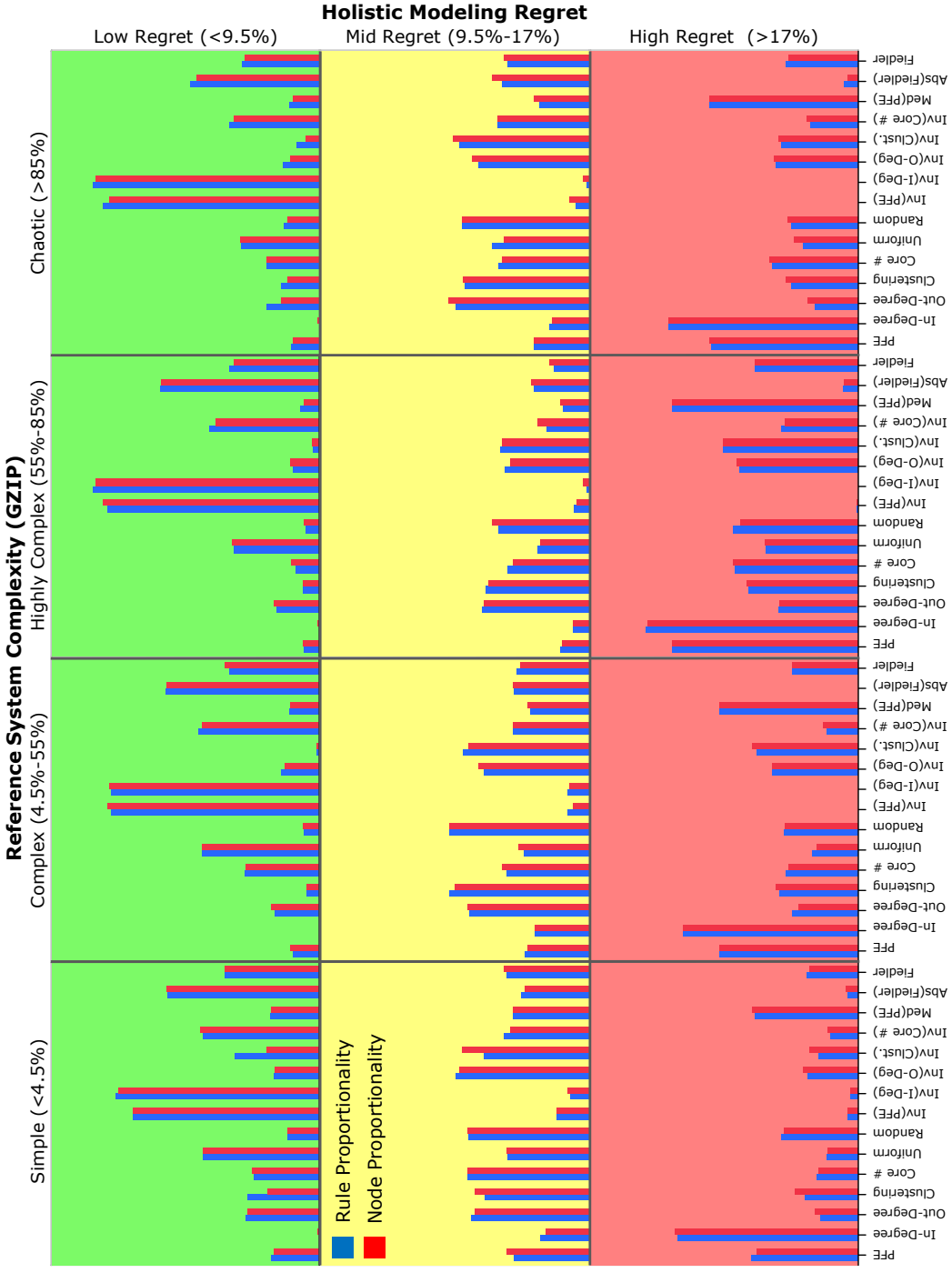

Figure 39: Holistic Modeling Regret for different node ranking schemes vs. Reference RBN complexity.

was tested for correlation for the 15 different ranking schemes. Figures 40 and 41 present the correlation between the schemes for 100 Random Networks (as created by the G(n,p) method) and another 100 Scale-Free Network (SFN)s (as created by the B-A Method). The top chart presents the percentage of cases where a p-value of less that 0.05 was identified, the bottom chart presents the average correlation identified. Note that the inverse rankings have a correlation of -1 with the proportional counterpart. Of interest is the high correlation between *In-Degree* and *PFE*, 91% for the SFNs and 92% for the $G(n, p)$. This difference is statistically insignificant, meaning that the rankings were correlated almost perfectly in both cases, and all 100% cases for both networks, a correlation significantly different from zero was identified. Other rankings that showed non-trivial correlations with PFE were the *Corrected Fiedler Vector* (inversely correlated) and the Core Number (proportionally correlated). Interestingly, *Inverse PFE* and *Absolute Fiedler* which performed fairly competitively, were completely not correlated. This may be an indication that a hybrid ranking scheme between the Fiedler vector and the PFE may provide the most suitable ranking for identifying the critical nodes since they operate on different information, but their independent ability to rank nodes is superior to most of the alternatives.

5.3.6 Test VI: Effect of nodes with zero ranking

The effect that the number of nodes with a ranking of zero have on the ranking scheme's ability to prioritize the nodes and model the behavior of the RBN was previously identified as a possible source of bias. For this reason, the relationship between the number of nodes with zero ranking and the modeling goodness was studied to see if there was a correlation. Figures 42 and 43 present the RMRs and HMRs for different amounts of nodes with null-ranking. The distributions do not indicate that having nodes with a ranking of zero affects the ability of that ranking scheme to prioritize the nodes and produce RBNs with high fidelity. In fact, in some cases, the opposite seems to be the case. *Corrected PFE* (Med(PFE)) has lower regret when more of its nodes have a ranking of zero. Other rankings do exhibit correlations that would indicate a negative impact from having nodes with a ranking of zero, e.g., *Proportional PFE* and *Proportional In-Degree*.

100 Erdős-Rényi Random G(n,p) Networks

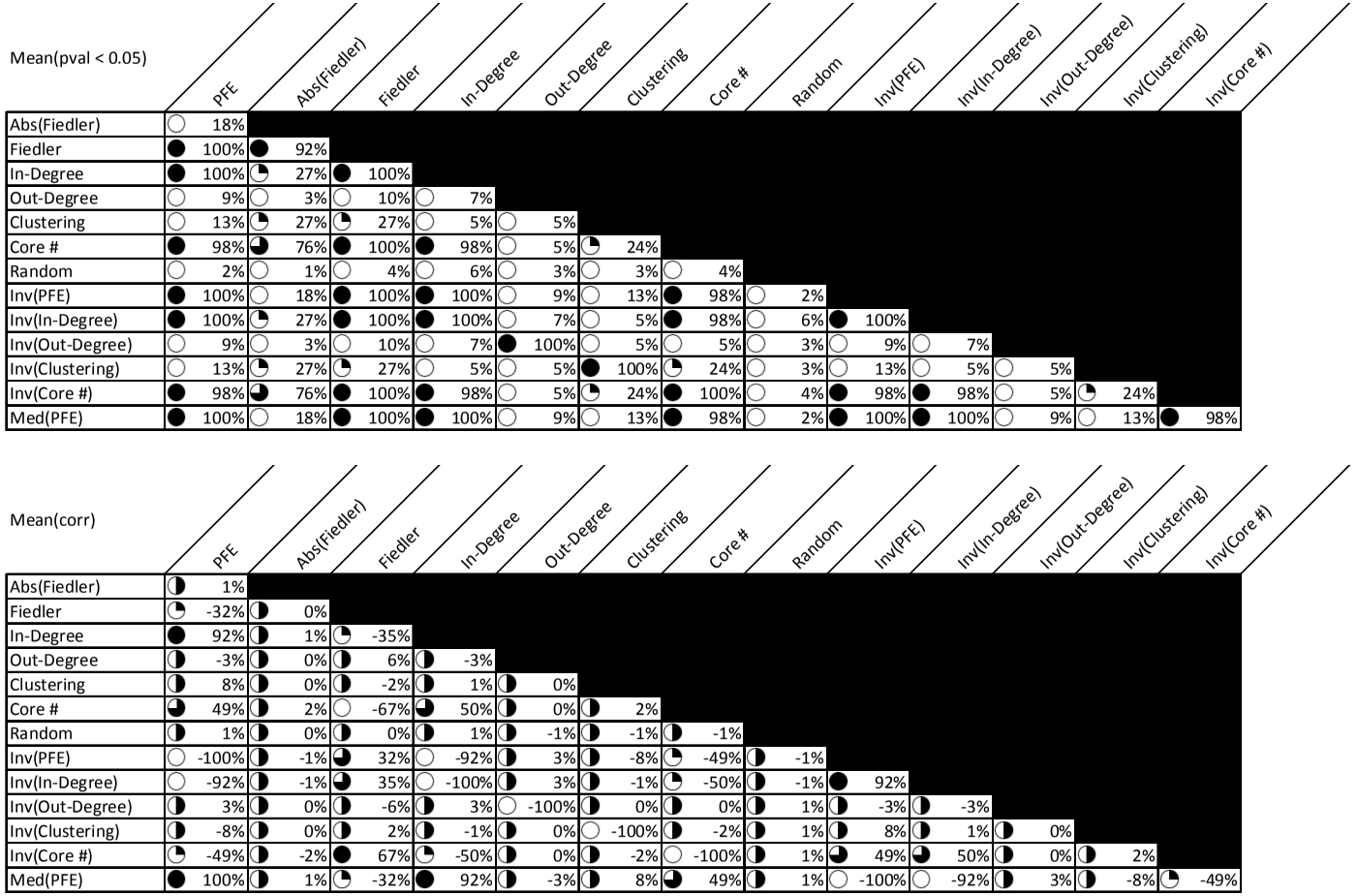

Figure 40: Node Ranking Scheme Correlations for Random Networks.

100 Barabási-Albert Random Scale-Free Networks

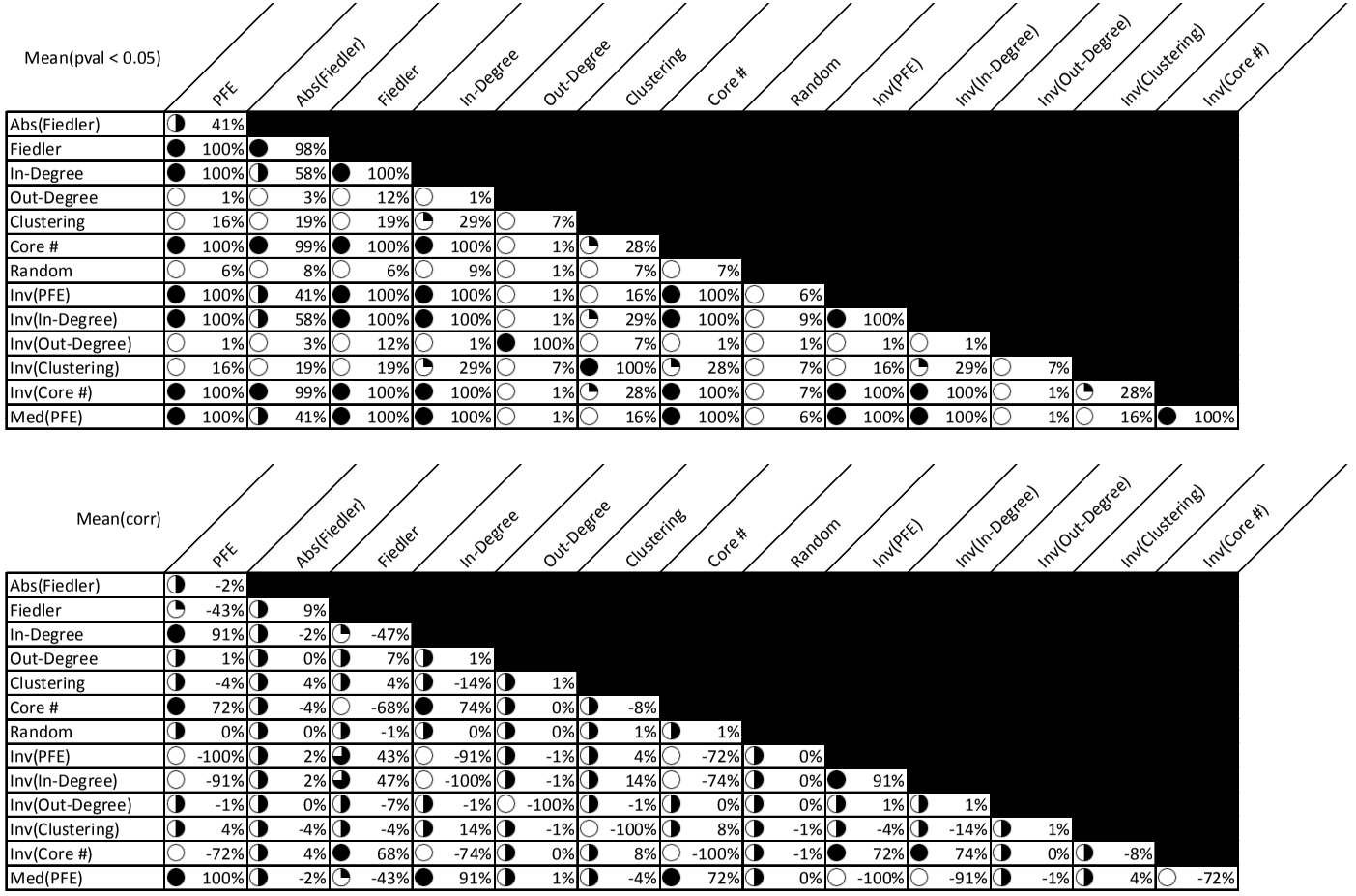

Figure 41: Node Ranking Scheme Correlations for Scale-Free Networks.

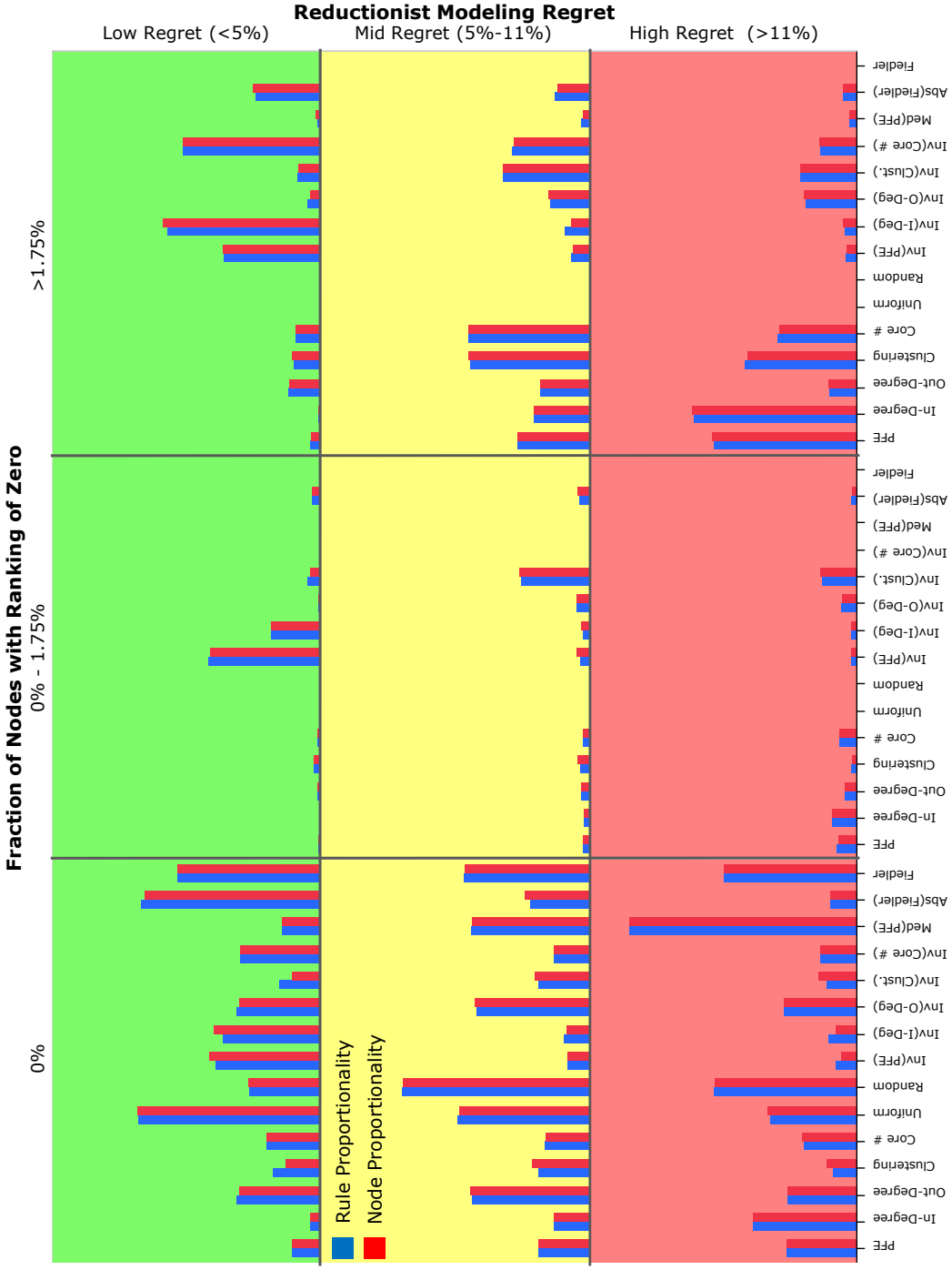

Figure 42: Effect of the number of nodes with ranking of zero on RMR.

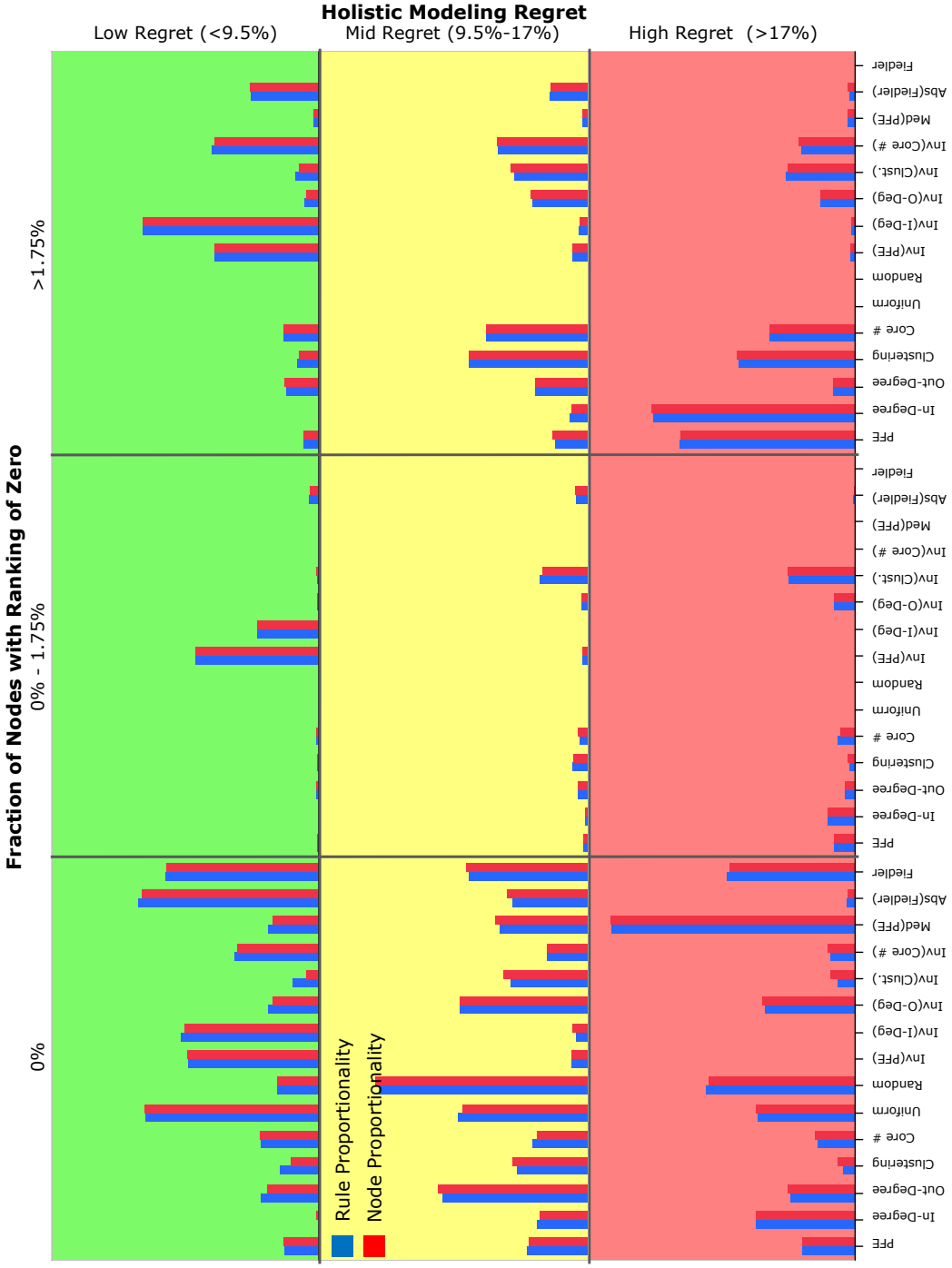

Figure 43: Effect of the number of nodes with ranking of zero on HMR.

5.3.7 Test VII: Rules Dismissal per Node

A possible reason why *Inverse In-Degree* may produce results that are considerably better than the uncorrelated rankings is that it is dismissing most of the rules of a handful of nodes, while other rankings are dismissing rules more equally. Since the number of rules per node is highly disparate (nodes with one input have 2 rules, while nodes with 16 inputs have 65,536 rules). This means that enough rules can be dismissed in only one node to satisfy the number required for that case. Table 14 describes an average of the average of number of rules dismissed for all the nodes. High values mean that most nodes had most of their rules dismissed, low numbers mean the opposite, that only a handful of nodes had most (or a fraction) of their rules dismissed. The critical finding of this test is that *Inverse In-Degree* and *Inverse PFE* for both Rule Proportionality and Node Exclusion dismiss rules in a significantly smaller number of nodes than the rest. The effect of this is that as the RBN evolves, the only rules that have been neglected are those in a few nodes, therefore, most of the RBN will behave as if nothing had changed, and will reproduce the behavior of the reference RBN. This also explains why *Proportional In-Degree* and *Proportional PFE* performed so poorly, these rankings neutralized the rules of most of the nodes, which in turn means that despite the fact that a few nodes still have most of their rules active, the behavior of the RBN will remain frozen and therefore emulate the behavior of the reference RBN poorly.

5.4 Alternative Rule Dismissal Experiment

In this experiment, the rules of the nodes were not dismissed by setting them to negative one. In fact, only one RBN was created. The modeling effort in this case was modeled as a probability that a given node at a given time step would not transition to the next state but remain in its given state. The probability that each node had of remaining unchanged was inversely proportional to its ranking. A parameter to control the number of cases where rules remain unchanged was used to ensure that the number of rules discarded could be set *a priori* as in the previous experiment.

The metrics for this second experiment were adapted slightly to produce more insightful

Table 14: Average rules dismissed per node.

Ranking	Rule Proportionality	Node Exclusion
$\mathbf{r}^{p(PFE)}$	94.6%	95.3%
$\mathbf{r}^{i(PFE)}$	1.5%	1.5%
$\mathbf{r}^{c(PFE)}$	94.5%	95.3%
$\mathbf{r}^{c(FV)}$	26.9%	27.1%
$\mathbf{r}^{a(FV)}$	20.4%	20.4%
$\mathbf{r}^{p(CC)}$	42.0%	42.3%
$\mathbf{r}^{i(CC)}$	51.1%	51.4%
$\mathbf{r}^{p(CN)}$	35.3%	35.2%
$\mathbf{r}^{i(CN)}$	28.3%	28.2%
$\mathbf{r}^{p(ID)}$	94.7%	95.3%
$\mathbf{r}^{i(ID)}$	1.3%	1.3%
$\mathbf{r}^{p(OD)}$	11.5%	11.5%
$\mathbf{r}^{i(OD)}$	49.5%	49.7%
$\mathbf{r}^{U}$	31.3%	31.2%
$\mathbf{r}^{R}$	36.7%	36.8%

results. The fidelity metric for the spectrum was no longer the integral of the difference between the RBN_R's spectrum and RBN_M's spectrum, but the R^2 of the natural logarithm of their amplitudes ($\ln a_R$ and $\ln a_M$, respectively)[12]. Equation 22 describes how this metric was calculated.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{f=0}^{f_{max}} \ln a_R - \ln a_M}{\sum_{f=0}^{f_{max}} \ln a_R - \text{mean}(\ln a_R)} \tag{22}$$

Furthermore, these R^2 fidelities were included in the reductionist fidelity category since the intent is to match the spectrum as closely as possible, not identify patterns in it. The holistic modeling metrics were also adapted slightly to condense their number. The holistic metrics studied in this secondary experiment are described below in Table 15. Where F_{PR} and F_{NR} stand for False Positive Rate and False Negative Rate respectively. The first three metrics describe how well peaks of the spectrum were identified, the last two metrics describe how well the superactive nodes and node correlations were identified. RMR is calculated as 50% of the binary regret plus 50% of the average of the regret of the R^2 of the decimal encoding, the Hamming distance, and the power of the TSM. HMR is calculated as 50% of the mean of DEC, HMD, and POW regrets, and the other 50% SAN and NC regrets.

5.4.1 Analysis of Results

> "If you are out to describe the truth, leave elegance to the tailor."
>
> - Albert Einstein

Figure 44 summarizes the results obtained in this second experiment. It represents the distribution of both RMR and HMR for different levels of complexity and modeling effort combinations. Unlike the bar charts used in the previous section, this time, the variability of regret will be represented using box plots. Box plots are represented as a box that

[12] Amplitudes of zero were neglected to avoid evaluating the logarithm of zero.

Table 15: Holistic Modeling metrics in Second RBN experiment.

Name	Description	Equation
DEC	Spectrum of the Decimal Encoding of the TSM	$\text{mean}(F_{PR}^{\text{DEC Peaks}}, F_{NR}^{\text{DEC Peaks}})$
HMD	Spectrum of the Hamming distance of the TSM	$\text{mean}(F_{PR}^{\text{HMD Peaks}}, F_{NR}^{\text{HMD Peaks}})$
POW	Spectrum of the Power of the TSM	$\text{mean}(F_{PR}^{\text{POW Peaks}}, F_{NR}^{\text{POW Peaks}})$
SAN	Superactive Nodes	$\text{mean}(F_{PR}^{\text{SAN}}, F_{NR}^{\text{SAN}})$
NC	Node Correlations	$\text{mean}(F_{PR}^{\text{NC}}, F_{NR}^{\text{NC}})$

range from the 25% to the 75% percentile, with a line through the center to represent the mean.[13] Figure 44 offers a large amount of information regarding the goodness of the different rankings, but this amount of information can be overwhelming, for this reason, a simplified version of these data is presented in Figure 45 where only the mean of the modeling regrets is presented. When looking at the means, the benefits of the *Proportional PFE* and *Absolute Fiedler Vector* rankings and the ranges for which these rankings are most suitable is clearly understood. As the reference system becomes more chaotic, the regrets are reduced since there is no clear ranking that offers and advantage over the others. Interestingly, for the most demanding case (chaotic system with least amount of modeling effort) *Uniform*, *Proportional Out-Degree*, *Inverse Core Number*, and *Corrected PFE* have the lowest RMR. In the case of HMR, the best rankings are *Corrected PFE*, *Inverse Core Number*, and *Uniform*. The fact that a *Uniform* ranking performs as well as it does indicates that in these extreme cases, ranking the nodes does not offer any advantage. If the effort is going to be distributed, in fact, this result shows that it is best to distribute the effort

[13] Symmetric distributions have the line in the center of the box, skewed distributions above or below it.

evenly amongst the nodes, and not randomly, in particular when creating holistic models.

For complex systems, ranking does offer advantages, in particular for RMR. In these cases, the *Absolute Fiedler Vector* ranking provides the minimal regret. Looking at the mean regret discards information on the robustness of the node ranking scheme. The benefits of rankings with smaller variability in regret can be appreciated if instead of studying Figure 45, one studies Figure 44. Study for example, the most demanding case, chaotic system with minimum modeling effort. By analyzing the mean, four rankings were competing for the minimum mean RMR. Figure 44 demonstrates that the *Corrected PFE* is better than the rest since its variability is considerably lower. This means that it is a more robust ranking for this condition. At the same time, for simple systems, where the best rankings were *Proportional PFE* and *Absolute Fiedler Vector*, the latter offers a significant advantage over the former, in that the *Absolute Fiedler Vector* has a much lower standard deviation for the minimal modeling effort case.

5.4.2 Reductionist Modeling

More information concerning the goodness of these node ranking schemes can be obtained if instead of analyzing the aggregated regrets, their individual regrets and original metrics are studied. Figure 46 represents the regret of the components of RMR. The four components of RMR are Binary, Decimal encoding, Hamming distance, and TSM power fidelities. The decimal encoding appears to have less variability than the rest in its regret. This is attributed to the encoding of the TSM, in that the large number of nodes, and their sorting based on their average state, produces time-histories that have small variability, with smooth spectrums. What is interesting is that the Hamming distance and Power fidelities have such high variability in regret (specifically for the simpler systems). In these simple systems with high modeling effort, *Proportional PFE* and *Absolute Fiedler Vector* based rankings produce the best results, and the best of the rankings depends on which of the sub-regrets one is interested in. If the focus is on Binary fidelity, *Absolute Fiedler* is the better ranking scheme. If the interest is on Hamming distance or Power, the *Proportional PFE* offers the best alternative. As modeling effort is reduced, *Absolute Fiedler* gains the

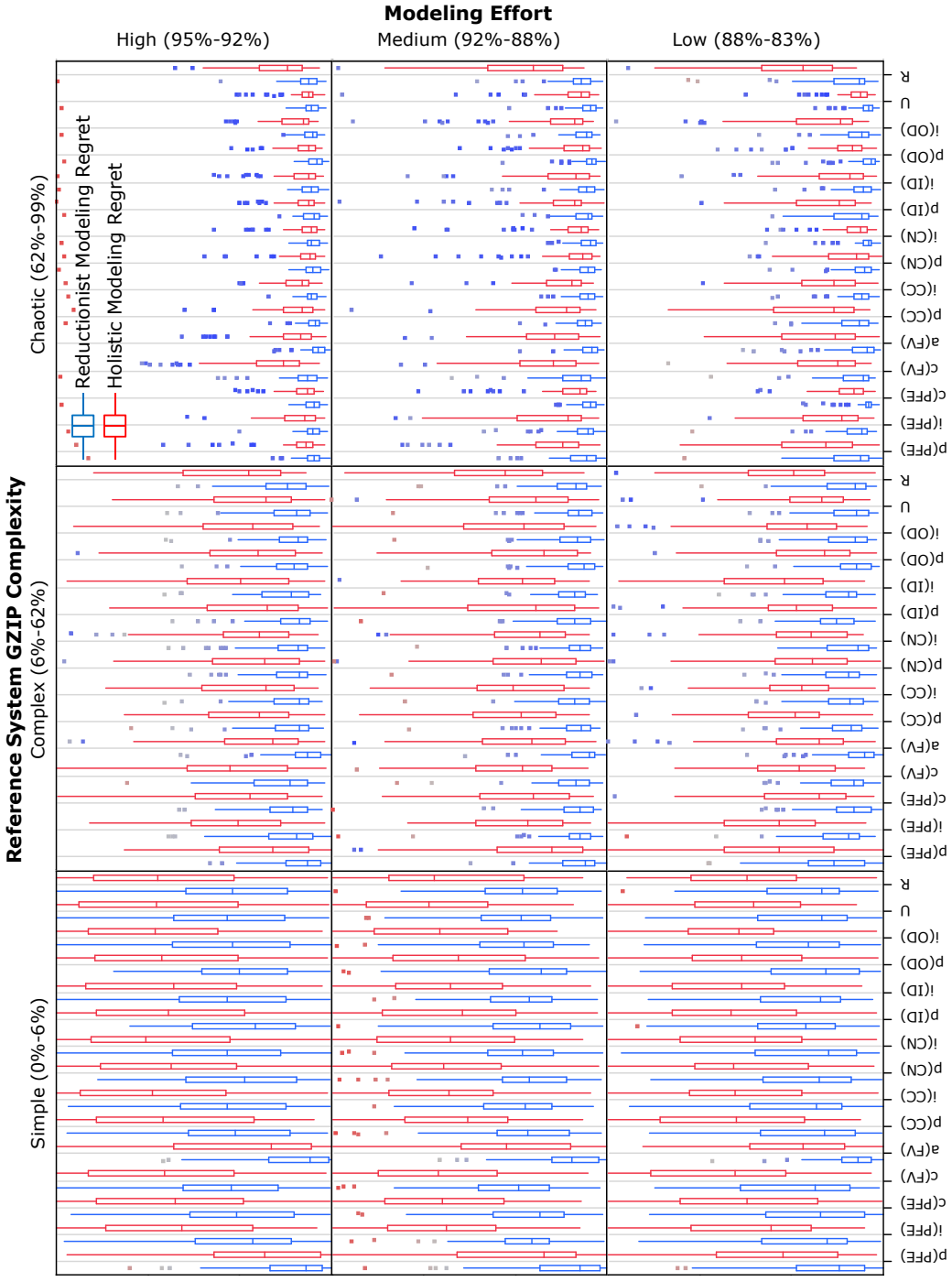

Figure 44: Reductionist Modeling Regret and Holistic Modeling Regret for the 15 node ranking alternatives vs. Reference system complexity and Transitions neglected.

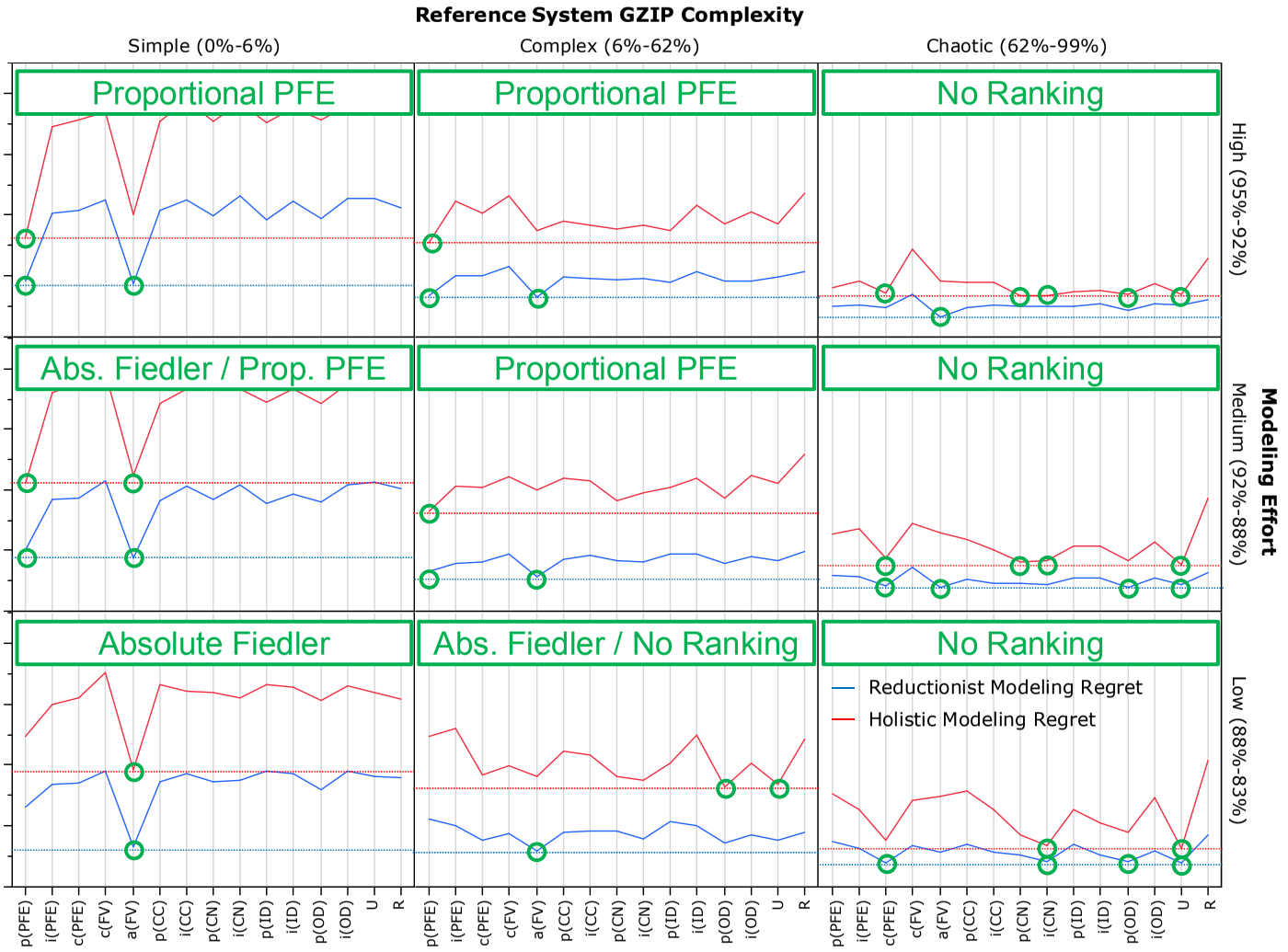

Figure 45: Mean Reductionist Modeling Regret and Holistic Modeling Regret for the 15 node ranking alternatives vs. Reference system complexity and Modeling effort.

advantage.

Analyzing regret simplifies the comparison, but confounds the absolute goodness of the alternatives. It may make small differences big by the way it normalizes, or it may make very bad alternatives seem appropriate when their absolute goodness is fairly poor. Figure 47 offers the four metrics that compose RMR by themselves ranked between zero and one, higher values are better in this case. Decimal encoding fidelity is in absolute terms, the best captured behavior for simple and complex systems. The interesting factor is that binary fidelity is captured better than the spectrum characteristics of the non-unique encodings of power and Hamming distance for these types of systems. It would be expected that the more demanding exact matching measured by the binary fidelity metric would be more difficult to maximize than the non-unique behaviors of the spectrum of Hamming distance and power.

An interesting but logical effect that can be discern from this visualization of the data is that as the system becomes more complex, the goodness of the different rankings reduces in variability for Hamming distance and power. For RBNs with simple behaviors, it is more difficult to pinpoint the spectrum of their Hamming distance and power, but as their behavior becomes more complex, and chaotic, their spectrums are easier to capture with fewer rules. Chaotic systems have noisier spectrums in these encodings and offer easier R^2 fits. The decimal encoding fidelity has for most of the rankings the most variability in the complex regime, the clear exception being the random ranking, which has the highest variability in the chaotic-low modeling effort regime.

5.4.3 Holistic Modeling

As demonstrated for RMR, HMR can be studied in more detail by analyzing the individual components of regret, and their absolute goodness. Figure 48 describes how the decimal encoding (DEC), Hamming distance (HMD), and power (POW) spectrum peak identification regrets, along with the with the node super-activity (SAN) and correlation (NC) identification regrets. In this case the regrets for the peak identification of the Hamming distance and power spectrums is highest for the simplest system and minimal when the system becomes

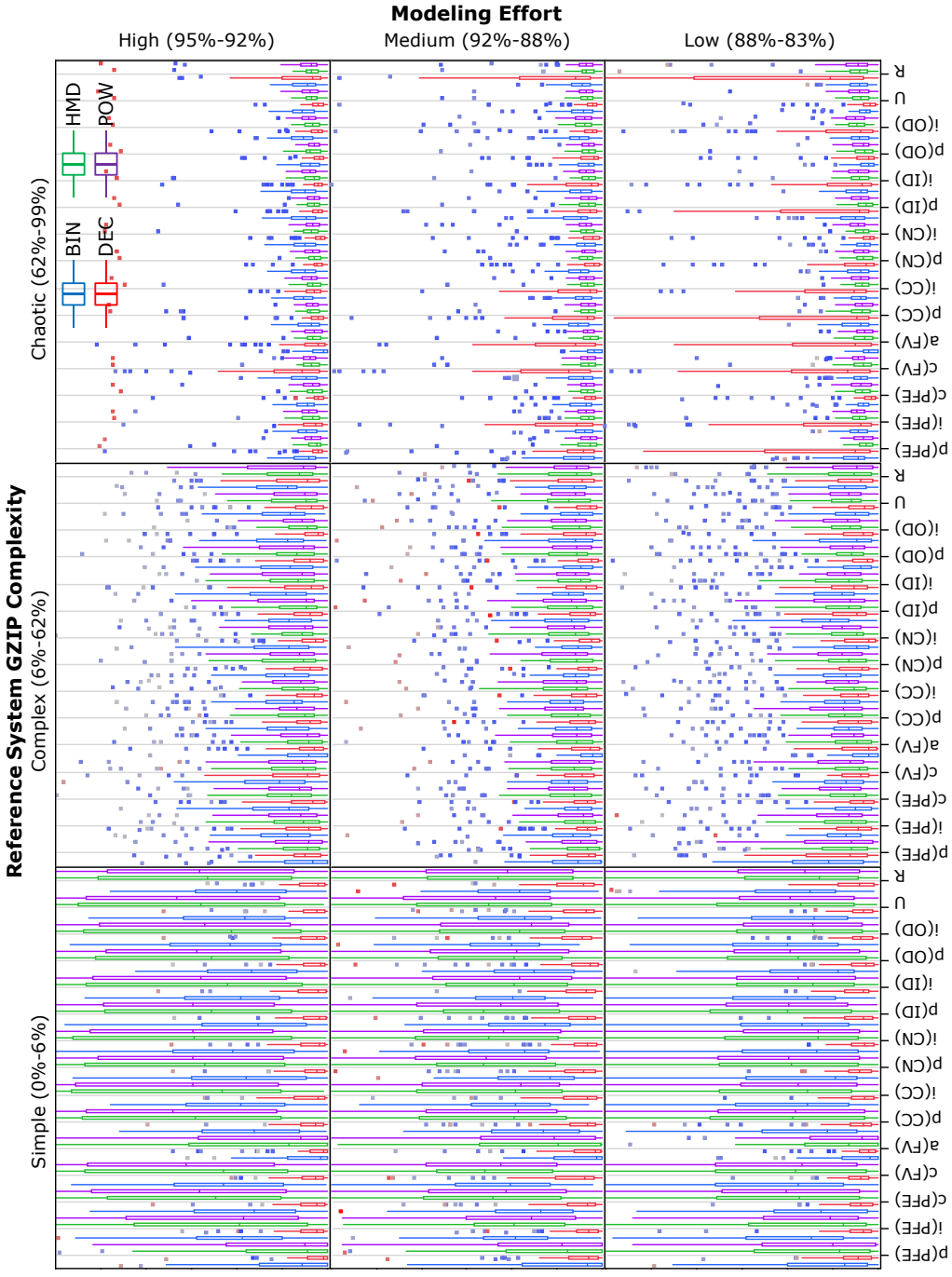

Figure 46: Components of Reductionist Modeling Regret for the 15 node ranking alternatives vs. Reference system complexity and Modeling effort.

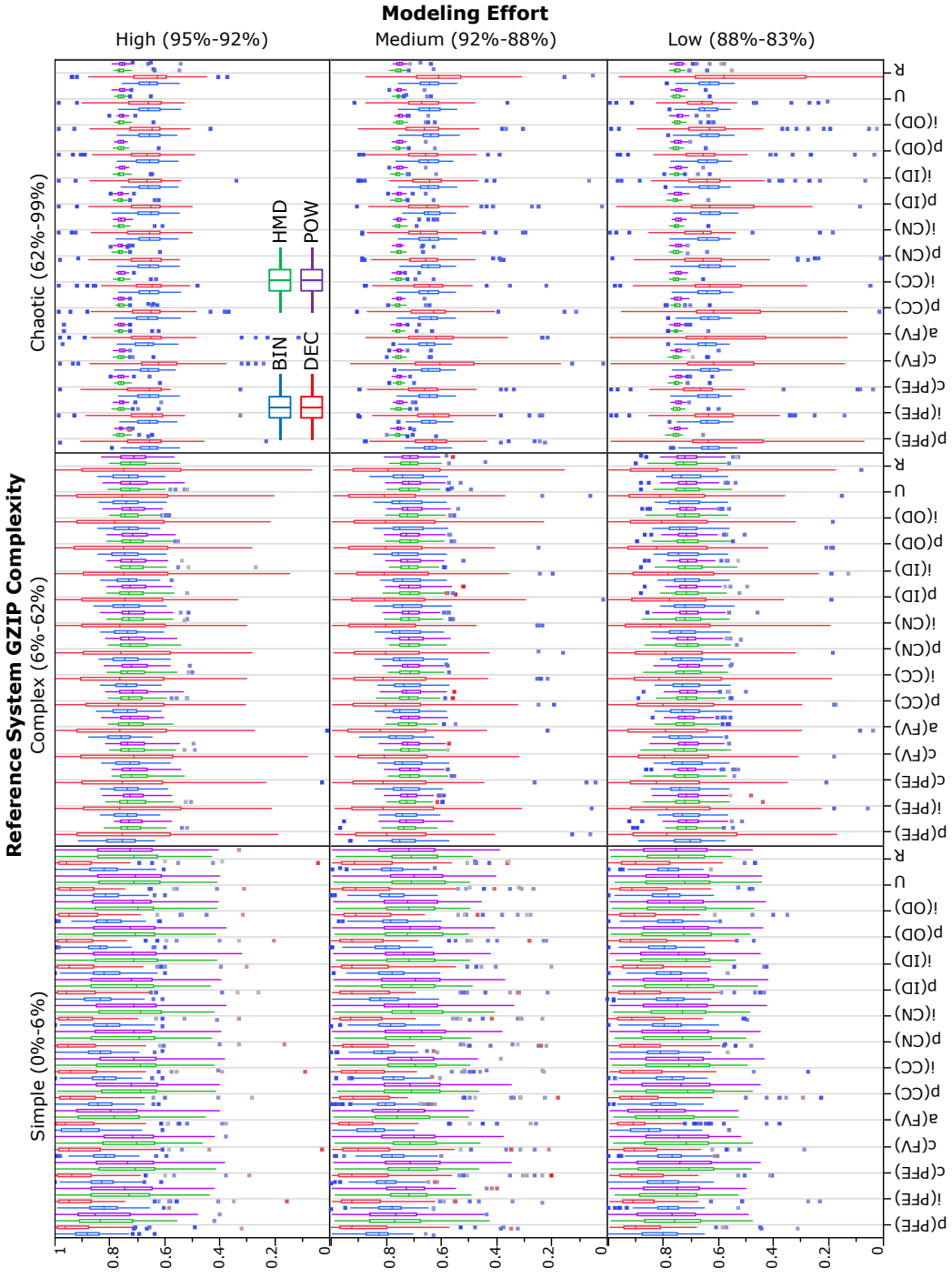

Figure 47: Reductionist Modeling Metrics for the 15 node ranking alternatives vs. Reference system complexity and Modeling effort.

complex. This is due to the same reason that was discussed previously for the spectrum fidelity. Node correlation and superactivity have the least amount of variability between the alternatives, and therefore, in general, the lowest regret. This means that as the system becomes more complex, in general, the hybrid HMR metric decreases in variability with the exception of the contribution of decimal encoding regret.

5.4.4 Ranking Scheme Goodness as a Function of Topology

The relative goodness of the different ranking schemes can also be studied as a function of topological characteristics of the RBN. This type of study provides guidelines that can be based solely on the structure of the complex systems and not its behavior (e.g., algorithmic complexity, etc.). Figure 50 presents RMR and HMR for various combinations of CNE[14] and the Digraph Algebraic Connectivity (DAC).[258] Of particular interest in these data is that the goodness of the ranking displays distinctive transitions depending on the combination of CNE and DAC. For low values of DAC, the *Absolute Fiedler* ranking offers the best alternative. For high CNE and high DAC, the *Proportional PFE* ranking offers the minimum regret. For intermediary combinations, it is either better to rank the nodes based on their absolute Fiedler value, or not rank them at all.

An alternative comparison between topological characteristics is between CNE and Off-diagonal Complexity (OdC). In Figure 51, the regret of the different ranking alternatives is compared for different combinations of CNE and OdC. The results indicate that for high values of OdC, ranking does not offer any benefit. For moderate values, the best ranking depends on the value of CNE. As seen previously, for higher CNE values, the *Proportional PFE* ranking offers the least regret, while for lower CNE values, *Absolute Fiedler* produces the best results. For low values of OdC, both *Proportional PFE* and *Absolute Fiedler* rankings are equally competitive.

[14] A normalized version of the cyclicity of the graph, i.e., $\lambda_1^{\mathcal{A}}/n$.

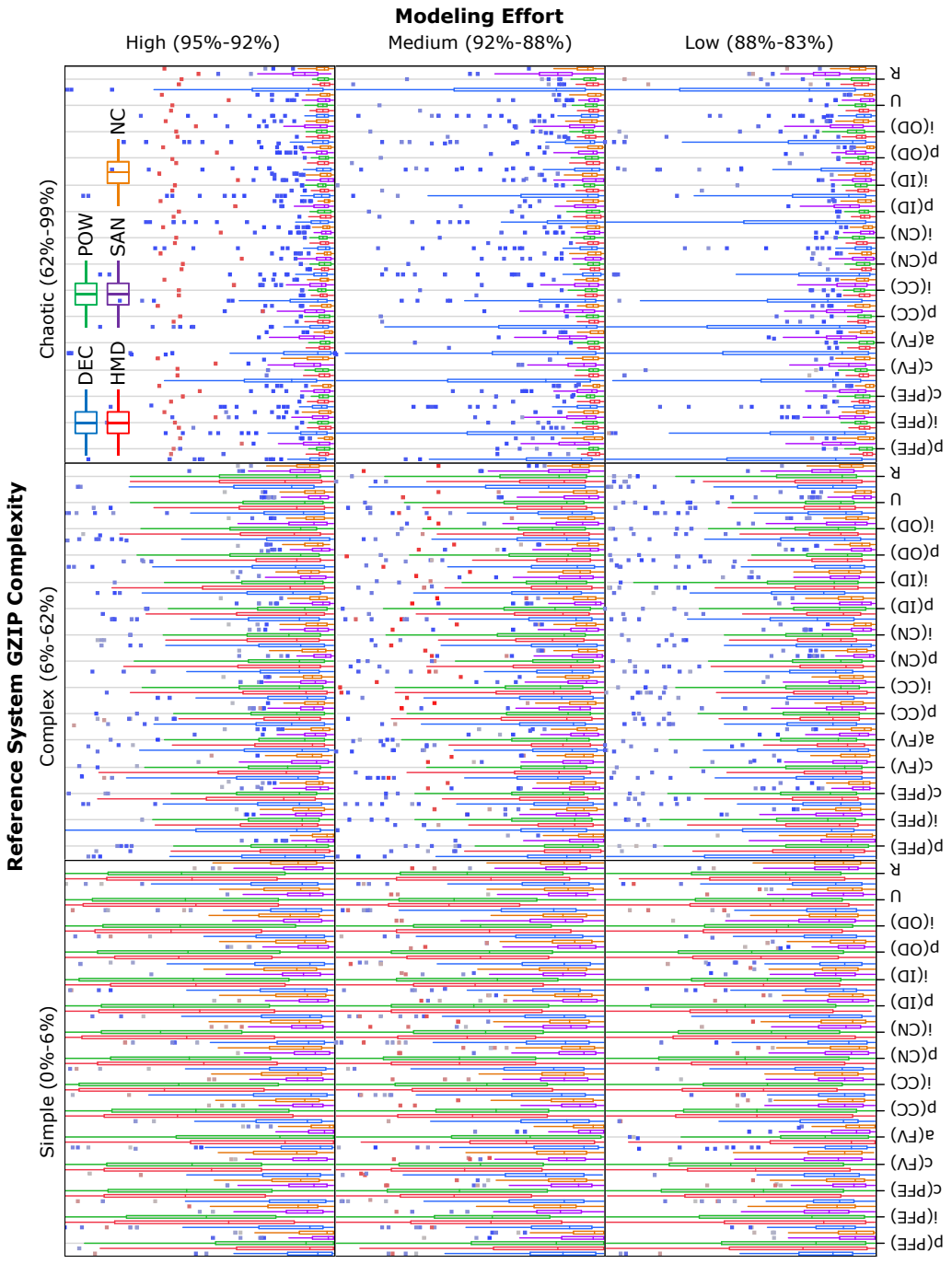

Figure 48: Holistic Modeling Regret for the 15 node ranking alternatives vs. Reference system complexity and Modeling effort.

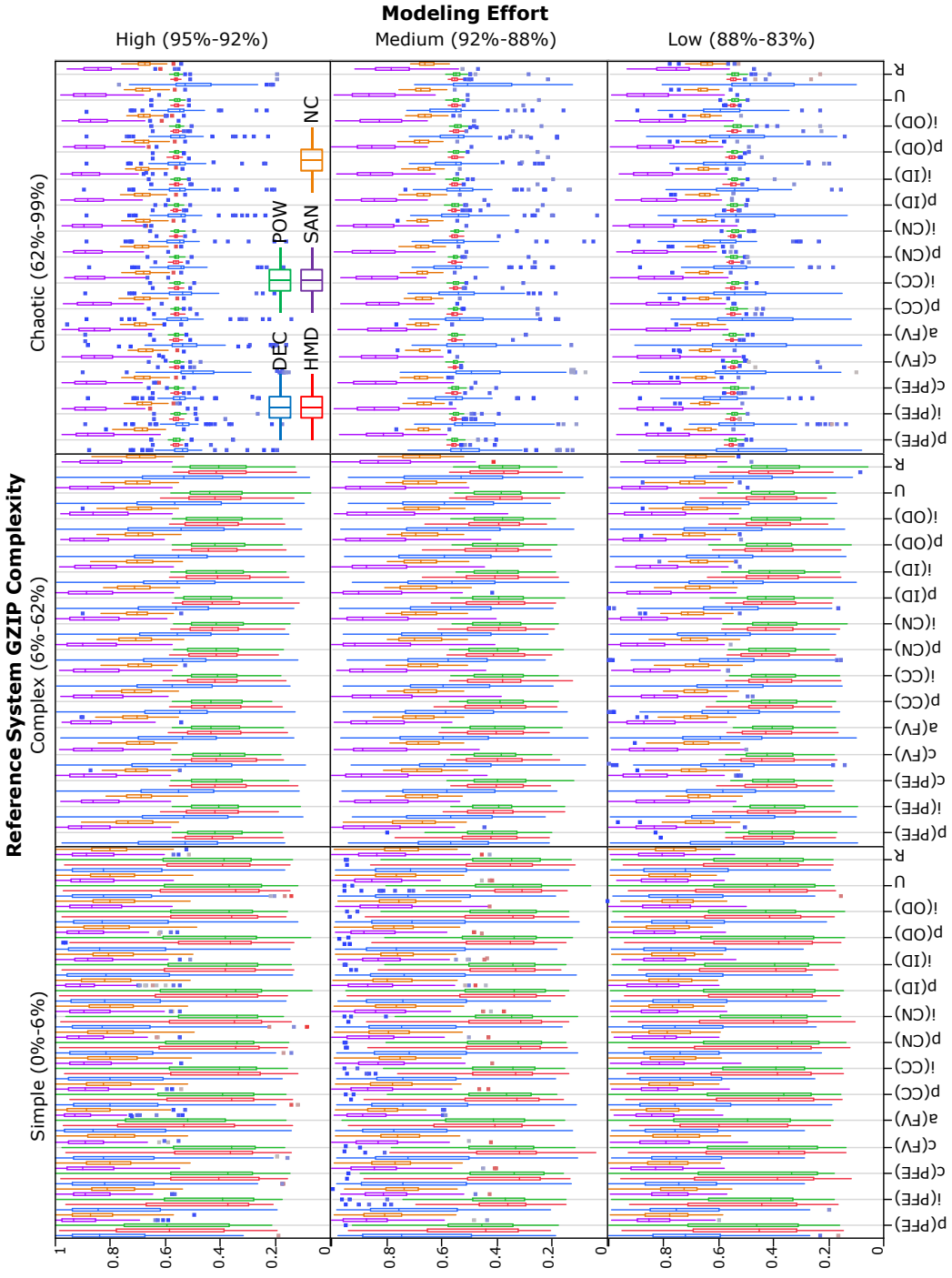

Figure 49: Holistic Modeling Metrics for the 15 node ranking alternatives vs. Reference system complexity and Modeling effort.

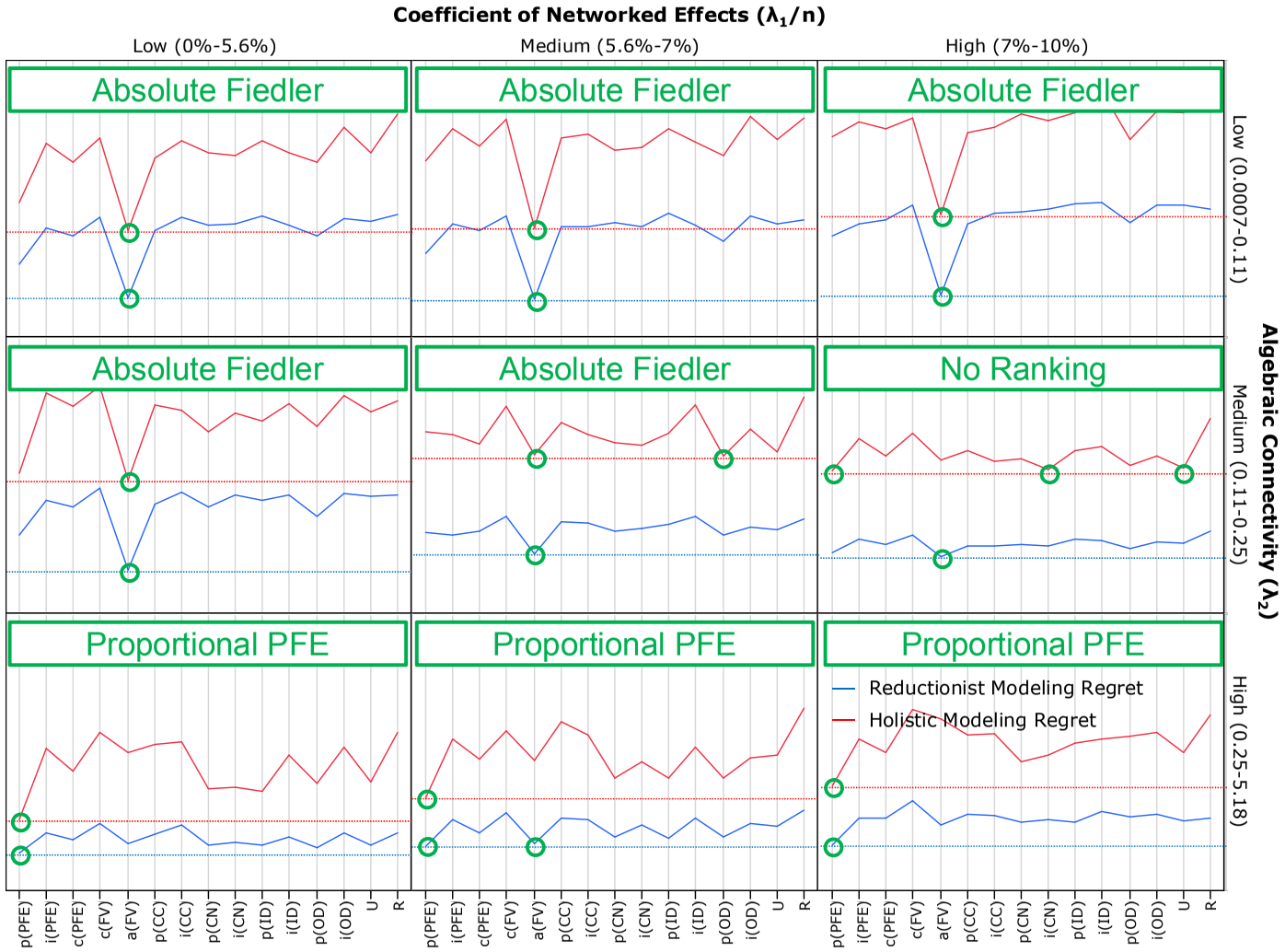

Figure 50: Mean Reductionist Modeling Regret and Holistic Modeling Regret for the 15 node ranking alternatives vs. Coefficient of Networked Effects and Digraph Algebraic Connectivity.

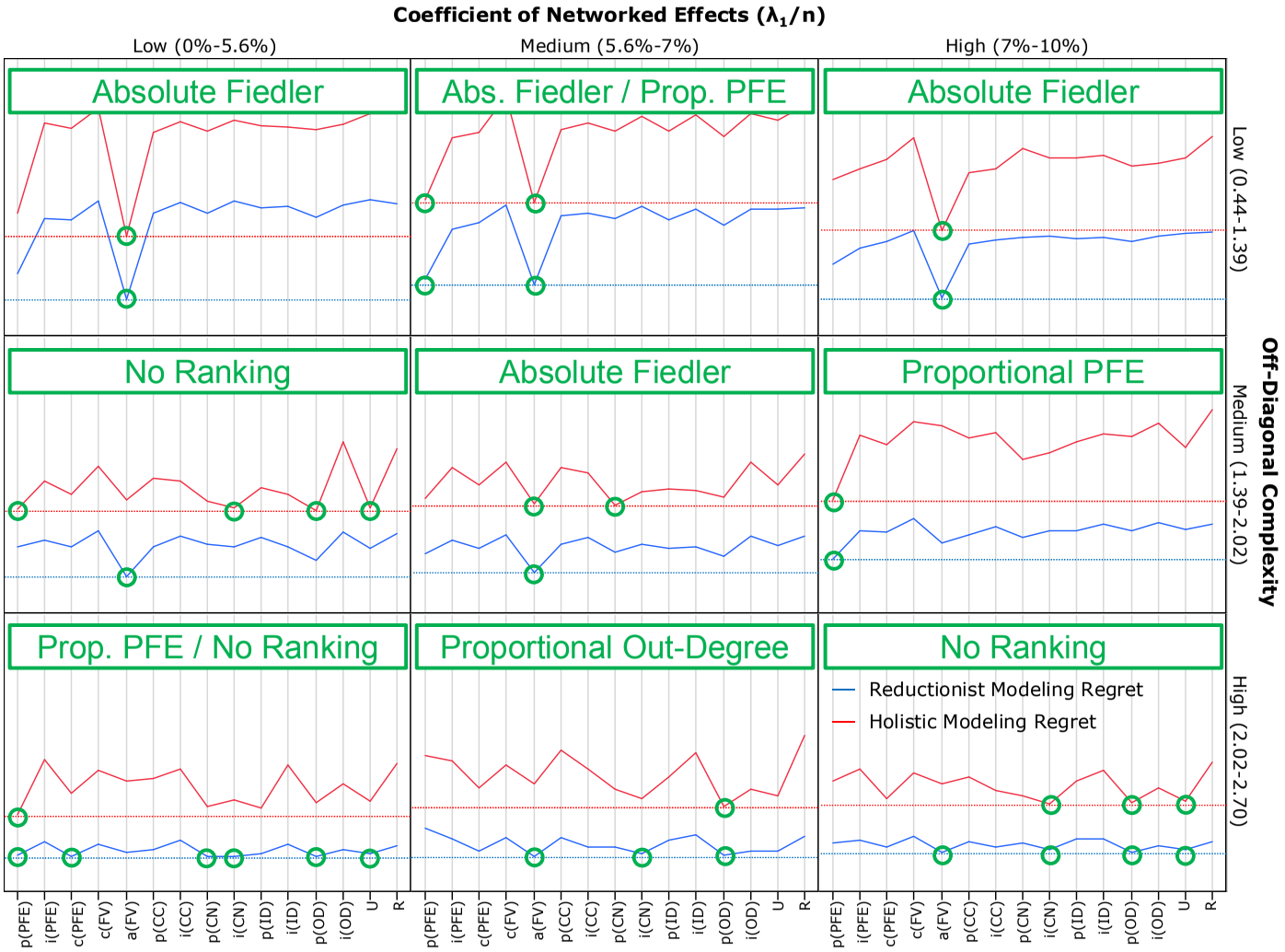

Figure 51: Mean Reductionist Modeling Regret and Holistic Modeling Regret for the 15 node ranking alternatives vs. Coefficient of Networked Effects and Off-diagonal Complexity.

5.5 Analysis Synthesis

If the individual rules of the nodes are to be dismissed as was shown in the first section, the best ranking schemes tend to "game the system" by eliminating only a fraction of the rules of a few nodes. Alternatively, other rankings do extremely poorly because they simplify the majority of the rules of the majority of the nodes. Useful insight was obtained from this portion of the analysis nonetheless, in that the probing into the reason to the goodness of certain rankings led to the study of their correlations and the identification of the *Absolute Fiedler Vector* ranking–that despite not concentrating the simplification on a handful of nodes–it performed competitively with the rankings that gamed the test.

The second test, which avoids having rankings "game the system" did in fact show that for certain types of systems, the regret of the PFE is lower, and therefore offers the best ranking of all the alternatives. Nonetheless, this was not the case for every system, depending on the structural metrics of the network, different rankings provided the best results. Three macro-level structural parameters of the network dictated which ranking is best, namely the OdC, CNE, and DAC. Figure 52 contains the 8 possibly combinations for the 3 metrics between their high (H) and low (L) settings. The first is the OdC, the second is the CNE, and the third is the DAC, e.g., the right middle box (HLH) is high OdC, low CNE and high DAC. As it was the case with the figures presented previously, the blue boxes represents the distribution of RMR and the red boxes the ones for HMR. The results presented in Figure 52 are synthesized in Table 16. The interesting feature, is that when both CNE and DAC have high values, it appears that the prioritization does not matter, regardless of the value of OdC. Of the 6 remaining settings, the *Proportional PFE* ranking provided the best results 3 times, the *Absolute FV* 2 times, and for the HLL setting, $\mathbf{r}^{a(FV)}$ offered the least RMR and $\mathbf{r}^{p(PFE)}$ the least HMR.

If modeling effort for the nodes are going to be prioritized, random prioritization is not a good approach. One result that all the figures and analysis in this section have supported is the fact that if the modeling effort is to be focused on some nodes, the selection of those nodes should not be done independently of the functional network of the macro system. Every piece of evidence observed in this section indicates that random ranking is worse

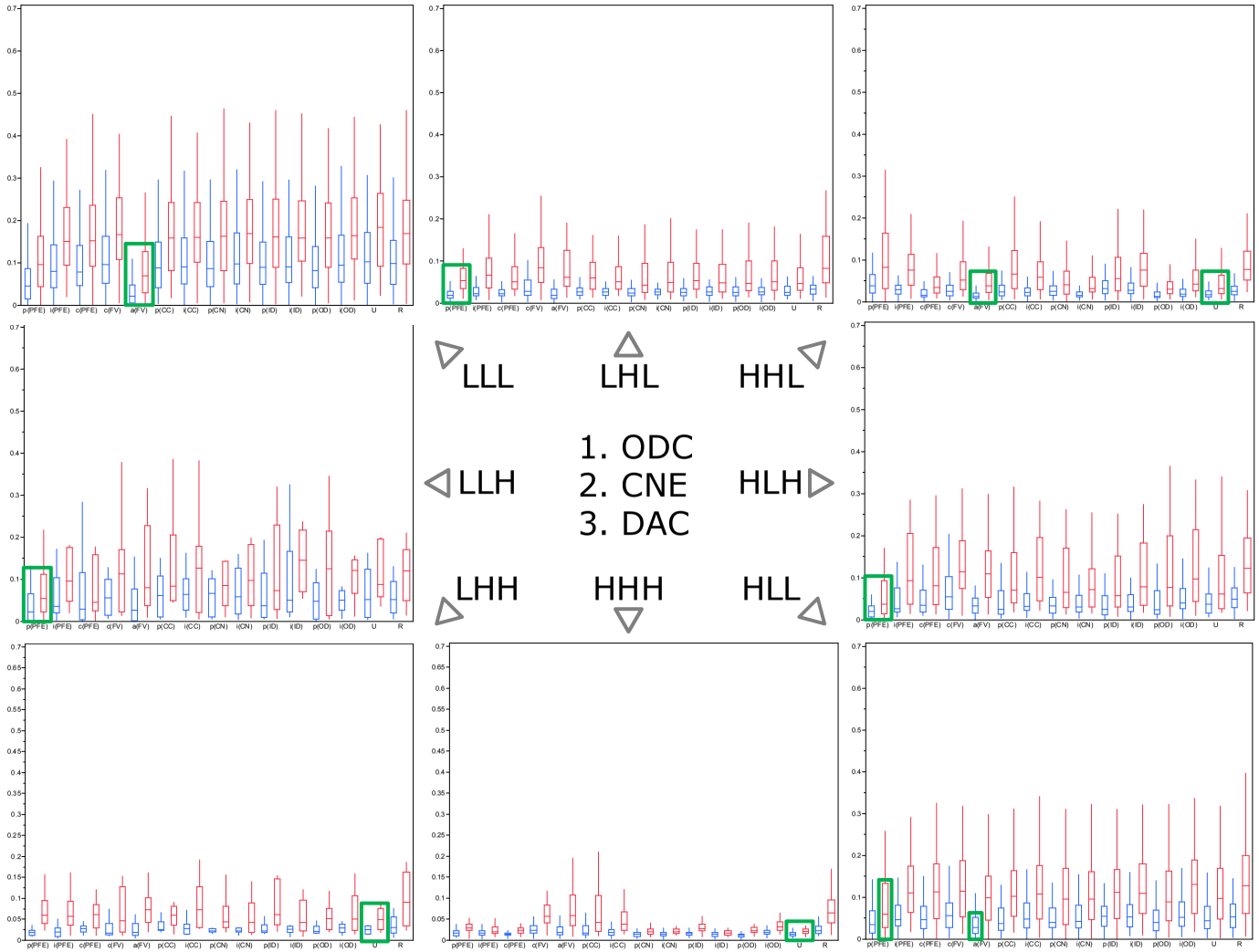

Figure 52: Reductionist Modeling Regret and Holistic Modeling Regret for the 15 node rankings vs. Coefficient of Networked Effects, Digraph Algebraic Connectivity, and Off-diagonal Complexity.

Table 16: Best Ranking Scheme vs. Network Structural Parameters.

OdC	**CNE**	**DAC**	**Preferred Ranking**
Low	Low	Low	$\mathbf{r}^{a(FV)}$
Low	Low	High	$\mathbf{r}^{p(PFE)}$
Low	High	Low	$\mathbf{r}^{p(PFE)}$
Low	High	High	No Prioritization
High	Low	Low	$\mathbf{r}^{a(FV)}$ or $\mathbf{r}^{p(PFE)}$
High	Low	High	$\mathbf{r}^{p(PFE)}$
High	High	Low	No Prioritization
High	High	High	No Prioritization

than uniform rankings, and generally much worse than other "intelligent" ranking schemes.

The approach taken to divide modeling regret into reductionist and holistic regret produced an unexpected result. In general, holistic modeling regret had a higher value, and a greater variability than reductionist regret.

For simpler systems, reductionist modeling produces lower levels of regret than holistic modeling regardless of which ranking scheme is used. Furthermore, two rankings produce clearly better results than the rest when RMR is concerned. As the system's behavior approaches the chaotic regime, the difference in RMR decreases. This is an indication that when the system's behavior approaches the chaotic regime, an intelligent approach to conceptual modeling provides no benefit over any other, or *over none at all*.

These tests were conducted with the goal of attaining statistical significance. Nonetheless, the portion of the possibilities studied is miniscule. For an RBN with 100 nodes, which is not a seemingly large network of boolean entities, there are 2^{100} different states that that RBN can have at any one time. Furthermore, there are 2^{9900} different topologies that network can have if one it so assume that every node is unique and the null graph and complete graphs are to be included. Furthermore, each node can have $2^{|inputs|}$ different rules, where $|inputs|$ is the cardinality of the input set (the number of nodes that affect that given node). These numbers are too large for most people to comprehend. As explained in Appendix B, a stack of 2^{100} pennies would span the universe 11/3 times, and in the entire universe, the best current estimate is that there are 2^{266} atoms. For this reason, statistical analysis may provide a rigorous test, but any of these computational results only represent a small fraction of all the possible results. Only an analytical approach can be used if these results are to be generalized to large-scale systems. The next section will focus on the analytical derivation of the importance of a node solely based on its relation to other nodes.

5.6 Analytical Study of Node Centrality

Let $\mathcal{D}$ be a directed graph composed of a finite set of vertices $\mathbf{V}$, and a set of ordered pairs (a, b) (where $a, b \in \mathbf{V}$) called edges $\mathbf{E}$, which are un-weighted, unique (no multiedges), and $a \neq b$ (no self-loops).

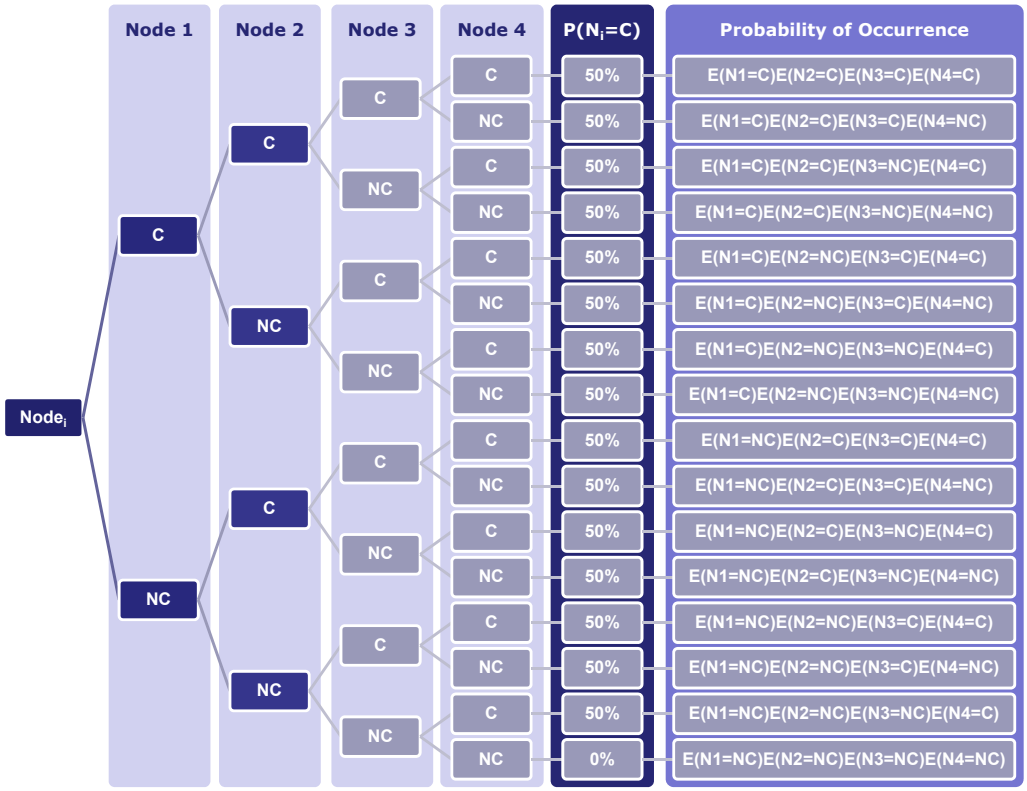

Figure 53: Probability tree for Node$_i$ to change state.

Let $\mathcal{A}$ be the adjacency matrix of $\mathcal{D}$, where

$$\begin{aligned} \mathcal{A}(i,j) &= 1 \text{ if } (i,j) \in \mathbf{E} \\ \mathcal{A}(i,j) &= 0 \text{ otherwise} \end{aligned} \tag{23}$$

Each node has a boolean state σ, where σ_i is the state of node i. At every time-step t, the state of a node is determined by a boolean function f. Each node has its own boolean function $f_i()$, the reader is reminded that there are $2^{2^{k_i}}$ different boolean functions, where k_i is the number of inputs to node i. For node i which has k_i inputs, its state at time $t+1$ is given by $\sigma_i(t+1) = f_i(\sigma_{j_1}(t), \sigma_{j_2}(t), ..., \sigma_{j_{k_i}}(t))$.

Now, the probability that node i will change state at time-step t is $P\left(\sigma_i(t) \neq \sigma_i(t-1)\right)$. Figure 53 illustrates the probability tree for node i to change state. In this example, node i is connected to four nodes. If none of the four nodes change state, node i has a probability

of changing state of zero. With any other combination, assuming a perfectly balanced rule set, node i has a probability to change state ($P(\sigma_i(t) \neq \sigma_i(t-1))$) of $\frac{1}{2}$.[15] Therefore, if node i has k_i inputs which had independent probabilities to change state $P(\sigma_j(t-1) \neq \sigma_j(t-2))$ at time-step $t-1$, then the probability that node i will change state, and the probability that it will remain in the same state are respectively given by Equations 24 and 25.

$$P(\sigma_i(t) \neq \sigma_i(t-1)) = \frac{1}{2}\left(1 - \prod_{j=1}^{k_i}(1 - P(\sigma_j(t-1) \neq \sigma_j(t-2)))\right) \quad (24)$$

$$P(\sigma_i(t) = \sigma_i(t-1)) = \frac{1}{2}\left(1 + \prod_{j=1}^{k_i} P(\sigma_j(t-1) = \sigma_j(t-2))\right) \quad (25)$$

In the initial iteration, each node that has a probability to change state equal to

$$P(\sigma_i(t=1) \neq \sigma_i(t=0)) = \begin{cases} 0 & \text{if } k_i = 0; \\ \frac{1}{2} & \text{if } k_i > 0. \end{cases}$$

Therefore, in the second iteration, each node has a probability to change state given by Equation 26.

$$\begin{aligned} P(\sigma_i(t=2) \neq \sigma_i(t=1)) &= \frac{1}{2}\left(1 - \prod_{j=1}^{\tilde{k_i}}\left(1 - \frac{1}{2}\right)\right) \\ &= \frac{1}{2}\left(1 - \prod_{j=1}^{\tilde{k_i}}\frac{1}{2}\right) \\ &= \frac{1}{2}\left(1 - \left(\frac{1}{2}\right)^{\tilde{k_i}}\right) \\ &= \frac{2^{\tilde{k_i}} - 1}{2^{\tilde{k_i}+1}} \end{aligned} \quad (26)$$

Where $\tilde{k_i}$ is the number of nodes adjacent to node i that have in-degree greater than zero. Figure 54 represents the probability that a node will change state in the second iteration based on the number of inputs with in-degree greater than zero. For completeness, the

[15] Proofs by Aldana [22] and Derrida and Stauffer [101], and Derrida and Pomeau [100] calculate macro-level probabilities of transition, this proof is concerned with node-level probabilities and relating those to the structure of the network without assuming constant in-degree as in Kauffman's original N-K RBN model.

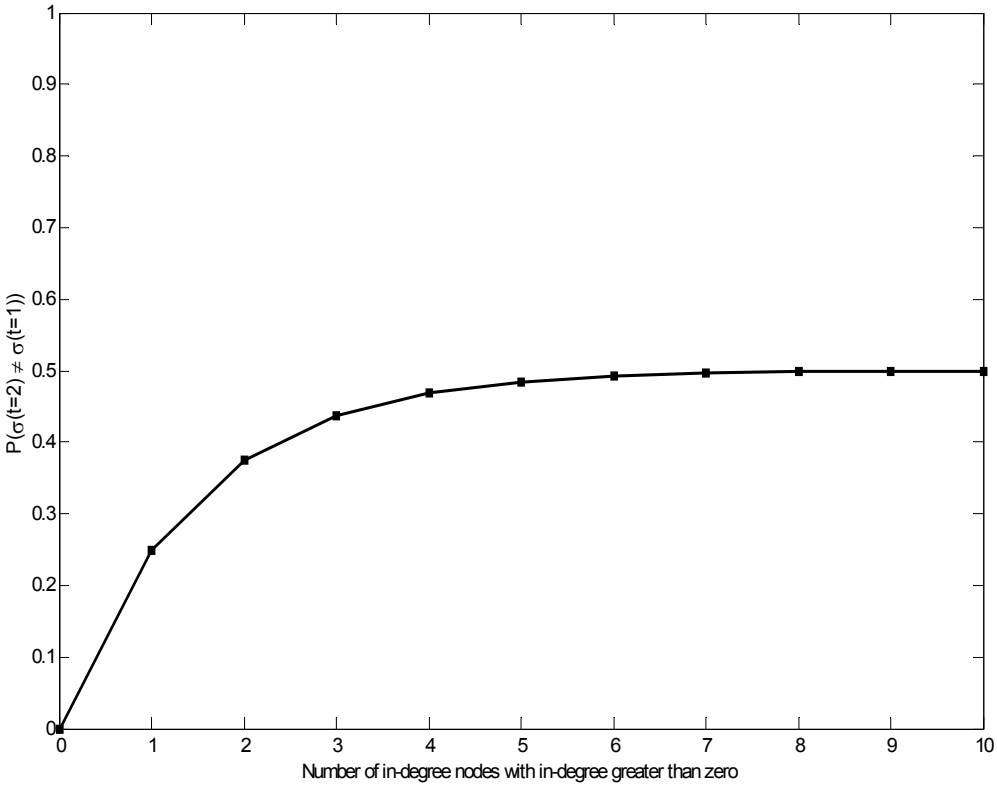

Figure 54: Probability for a node to change state in the second time-step as a function of its in-degree.

probability that a node will remain in the same state in the second iteration is given by Equation 27.

$$P\left(\sigma_i(t=2) = \sigma_i(t=1)\right) \quad = \quad \frac{2^{\tilde{k_i}} + 1}{2^{\tilde{k_i}+1}} \tag{27}$$

On time-step 3, $t = 3$, the probability that node i changes state is given by

$$P\left(\sigma_i(t=3) \neq \sigma_i(t=2)\right) = \frac{1}{2}\left(1 - \prod_{j=1}^{\tilde{k_i}} \frac{2^{\tilde{k_j}} + 1}{2^{\tilde{k_j}+1}}\right) \tag{28}$$

Equation 28 has no simple analytical solution. If we assume that the adjacent nodes to node i have equal non-trivial in-degrees–i.e., constant $\tilde{k_j}$ for node i–then the probability that node i will change state in the third time-step is given by Equation 29. Figure 55

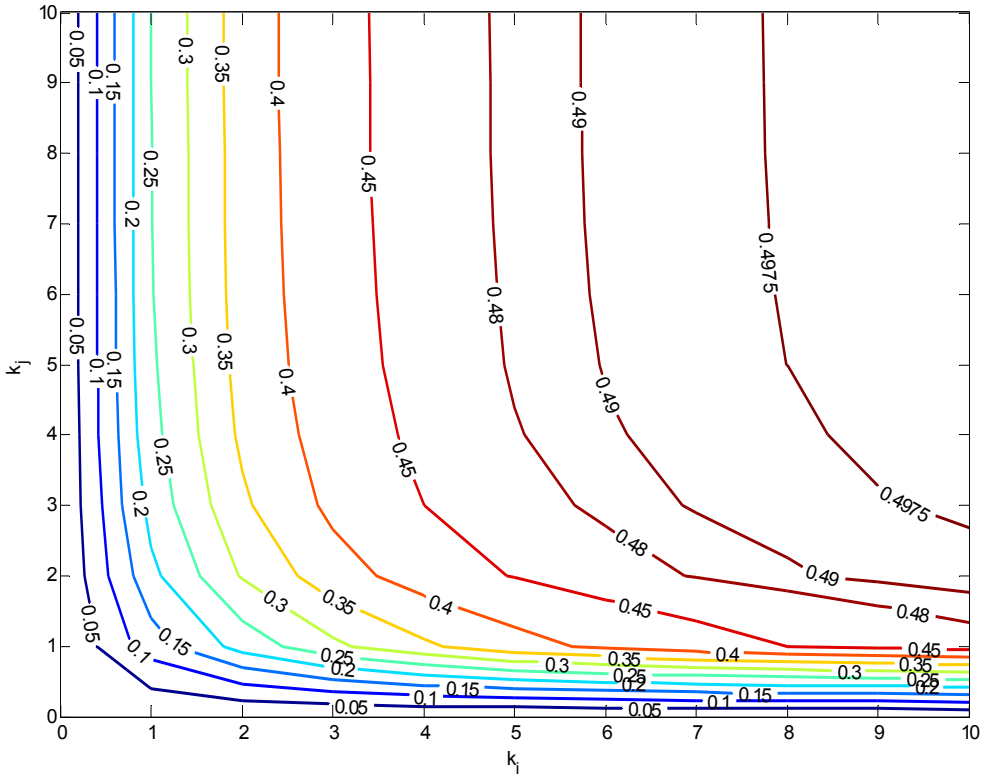

Figure 55: Probability for a node to change state in the third time-step as a function of $\tilde{k_i}$ and $\tilde{k_j}$.

graphically displays this probability as a function of $\tilde{k_i}$ and $\tilde{k_j}$. From the figure, it is clear that the behavior is driven by $\tilde{k_i}$, where, as long as $\tilde{k_j}$ is greater than zero, the probability that a node will change state is dominated by the number of nodes adjacent to i.

$$P\left(\sigma_i(t=3) \neq \sigma_i(t=2)\right) = \frac{1}{2}\left(1 - \left(\frac{2^{\tilde{k_j}}+1}{2^{\tilde{k_j}+1}}\right)^{\tilde{k_i}}\right) \tag{29}$$

In conclusion, this analytical analysis of Random Boolean Networks shows that in-degree dominates the probability a node will change state, and the in-degree of the adjacent nodes, henceforth, has an effect as well. A promising avenue of study is to create a proof based on Markov Chains to study the probability of a transition taking place. This will be left as future work.

5.7 Application of Hypothesis B to gDNOsim

This section will illustrate how the lessons learnt in the testing of Hypothesis B, can be applied to a practical M&S problem and integrated with the lessons learned in the testing of Hypothesis A. In this section, this process is conducted for one specific problem, and is not meant to be a rigorous test of neither hypothesis. The author would like to reemphasize that the purpose of this thesis is not to develop a method, but to test a series of hypotheses. In this application, the process developed to test Hypothesis A, acDiMA, will be used as the basis for generating representative networks from which the prioritization of the species can be done.

5.7.1 A process to rank agents from DiMA products

The lessons learned in the testing of hypothesis B can be integrated with the Digraph Modeling for Architectures technique described in Section 4.1. DiMA generates functional graphs that are representative of those that would be observed when the architecture is fulfilling its intended capability. The node rankings studied in this section and their ability to identify critical entities can be used to identify the critical nodes in the architecture from the same products that DiMA uses to study the functional cyclicity of a large-scale system architecture. Depending on the observed CNE, DAC, and OdC, the modeler can then study different node ranking metrics computed from the engagement matrices created by DiMA and determine which agent types are more central to the overall behavior of the architecture.

The process used to demonstrate how this integration can take place and illustrate the validity of the outputs is presented in Figure 56. The nine steps followed by this process are described below. In essence, this process constructs an ABM model in gDNOsim, perturbs that model 20 times by changing the characteristics of each species to obtain different variations in the metrics tracked. For each of these 20 variations, their difference with the results of the reference model is computed. In parallel, a DiMA analysis is computed based on the characteristics of the reference run of gDNOsim and the engagement matrices from DiMA are used to compute the PFE and FV rankings for the 20 species. All these steps are

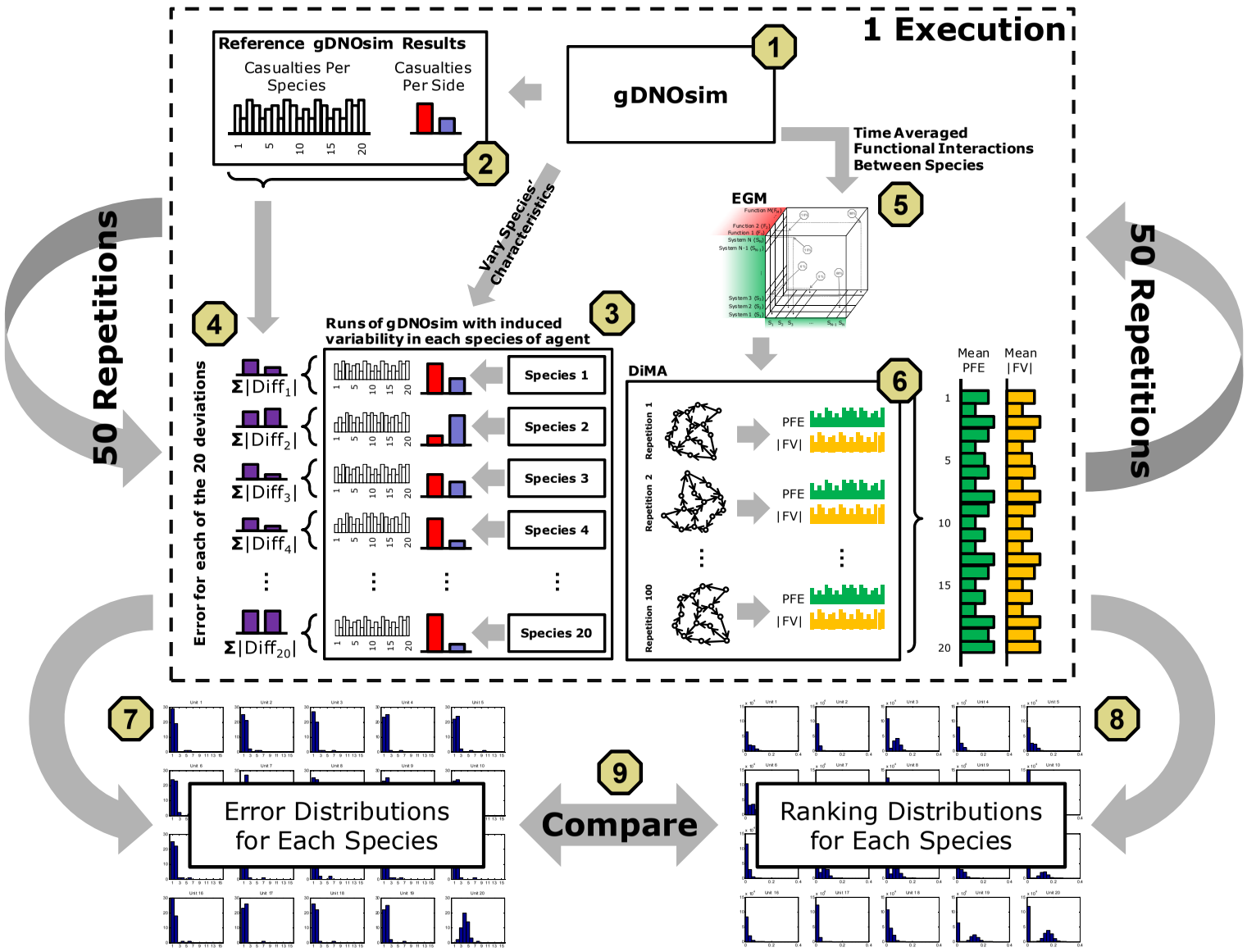

Figure 56: Test of ABM application of Hypothesis B.

Side	NetLogo ID	Number of Type	Speed	Max-Turn	RCS	Sensor Range	Sensor Angle	S/N Required	Time-to-Track	Ammo	Shoot Range	Min Time to Lock-On	Comm Range	Decider
Blue	1	11	0.46	14.4	6.37	3.3	261	2.06	14.9	11	10.8	5.4	13.0	F
Blue	2	22	0.34	9.2	8.63	20.2	20	4.48	14.4	7	19.4	18.5	73.8	F
Blue	6	14	0.21	6.5	1.31	0.0	36	8.68	14.4	11	1.1	9.1	74.4	F
Blue	7	13	0.17	12.6	8.02	20.9	312	1.44	9.7	10	2.7	13.4	9.0	F
Blue	9	25	0.36	11.2	3.56	25.7	88	2.71	6.8	10	4.9	4.0	33.4	F
Blue	10	15	0.64	13.0	2.60	33.2	178	8.92	7.7	15	19.4	18.9	16.7	F
Blue	11	20	0.52	12.8	4.30	16.2	119	2.42	14.9	9	14.8	18.8	57.6	F
Blue	13	13	0.20	5.9	7.28	16.0	216	4.92	7.5	2	4.1	8.4	41.2	F
Blue	14	13	0.24	12.0	6.68	14.5	349	7.34	6.1	6	5.6	10.0	30.0	T
Blue	16	14	0.52	6.6	1.07	15.8	263	1.52	10.0	2	11.6	3.9	10.1	F
Blue	17	18	0.18	5.6	6.92	18.2	94	5.16	14.8	4	1.2	7.4	38.3	F
Blue	18	13	0.50	10.2	6.44	5.1	354	7.71	12.6	10	5.8	5.3	37.1	T
Red	0	11	0.83	14.1	2.01	16.4	105	8.19	12.4	1	11.5	5.2	71.7	F
Red	3	12	0.38	5.2	5.65	23.7	102	1.91	7.7	13	9.8	7.8	23.8	F
Red	4	14	0.83	12.2	8.70	27.7	247	4.89	9.1	15	14.4	10.1	31.6	F
Red	5	19	0.40	6.1	7.31	31.5	209	2.42	6.1	7	10.8	17.0	40.3	F
Red	8	15	0.16	8.0	1.32	35.8	134	8.68	8.7	9	5.9	3.9	44.2	F
Red	12	21	0.77	5.4	2.22	2.9	49	6.06	7.1	4	9.5	6.4	70.7	F
Red	15	11	0.39	10.9	3.75	15.8	330	3.90	10.8	12	17.8	10.2	20.7	F
Red	19	24	0.63	9.1	1.01	13.6	44	6.19	14.8	7	18.8	15.6	61.3	T

Table 17: Unit types in the gDNOsim example for Hypothesis B.

repeated 50 times to obtain statistically significant results. These results from the rankings and the imperfect ABSs are then compared to study the ability of the rankings to predict the "criticality" of each species.

1. The first step in the test is to identify a reference model which serves as the analogy for reality. In this case, the reference model will be an instantiation of a gDNOsim[16] model, with 20 different species of agent. The description of the 20 species is presented in Table 17. In summary, 12 of the 20 species belong to the blue side, and 8 to the red. Blue has two decider species and red has one, the colors for the unit characteristics are color coded, green being the best, red the worst, e.g., for RCS low is green, high is red, but for ammunition it is the opposite.

2. The second step consists of obtaining metrics from the execution of gDNOsim. In this case, two metrics will be tracked: (1) fraction of casualties per side, and (2) fraction of surviving species. These metrics are not the only possible metrics, but for illustration purposes, they are representative of what a force attrition exchange analysis would involve. The simulation is stopped when there are no casualties recorded for 150 continuous time steps.

3. In the third step, each of the species in the gDNOsim model is modified slightly to induce some variations to their characteristics. This is representative of modeling them incorrectly, of having an error in the representation of the agents. These deviations of the reference model are analogous to the *models of the model* described in the previous sections for the Hypothesis B tests using the RBN formulation. Since there are 20 species, 20 separate deviations of the reference model are executed.

4. For each of the 20 deviations, the two metrics tracked for the reference model are recorded and the difference between each one of them and the results from the reference model are then computed. This difference (or error) is computed as the summation of the absolute difference. Models of the model that match the reference model have

[16]For more information on gDNOsim the reader is referred to Section 4.3.3.

smaller error.

5. The reference run of gDNOsim is used to obtain time-averages of the functional interactions between all species to construct an accurate EGM. This step is to ensure, as it was the case in the testing of DiMA, that the DiMA process uses accurate representations of "reality" by obtaining accurate representations of the reference model's functional interactions.

6. The EGM obtained from gDNOsim is then used by DiMA to generate 100 different possible engagement matrices. The PFE and the absolute value of the FV are then computed for the 100 different engagement matrices, and the distributions of PFE and absolute value FV are then coalesced for the 20 different species.[17]

7. Steps 1 through 6 are repeated 50 times to obtain statistically significant results since the simulations conducted are stochastic. The error distributions for each of the 20 deviations of the gDNOsim are then collected for the 50 repetitions. The result is a distribution of error for each of the 20 species.

8. The PFE and absolute value FV rankings for each of the 50 repetitions is collected. The result is a distribution of ranking importance for each of the 20 species.

9. The last step consists of comparing the distributions of error and importance. If the process helps identify the critical species, then the species that produced the largest error when its characteristics were varied, should receive the highest rankings.

As explained previously, DiMA produces networks that can be analyzed in the same fashion that the RBN networks were analyzed to identify the critical entities. The value of Hypothesis B is that it relates the structure of the system to its behavior, and stipulates a relationship between the centrality of the nodes of a system and their impact on the overall system's behavior. The engagement matrices generated by DiMA, the matrices

[17]Since there are multiple agents for each species, and they have a different entry in each computation of the PFE and FV, their values must be statistically coalesced (in this case by summation) to obtain 20 values of PFE and FV.

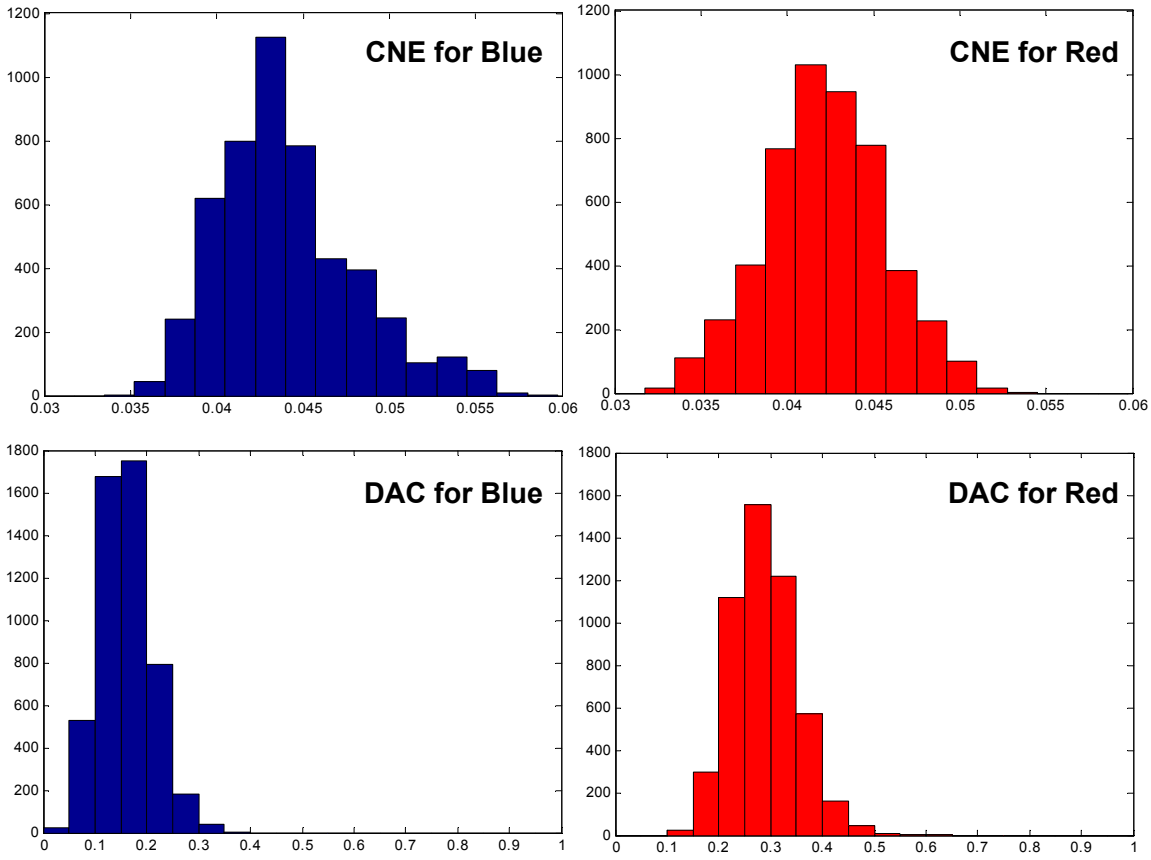

Figure 57: CNE and DAC for the Blue and Red capabilities in the ABM application of Hypothesis B.

that represent possible engagement scenarios, can serve as the networks from which the node rankings are to be derived from. There is one consideration that must be specifically addressed. These networks are not between species, but agents specifically, where there may be multiple agents of the same species, but since the EGM is a probabilistic matrix, only a subset of the possible functional relations will be active. That means that every agent within a species is likely to receive a different value by the rankings. There are multiple ways in which the rankings for the entire species could be computed. A geometric mean would neglect the number of agents in a given species by normalizing according to their individual rankings and their numbers. A summation would serve the same purpose but linearly account for the number of agents in every species. Recording the maximum value for each species is a third alternative, but as it was the case with the mean, it does not directly account for the numbers in each species. In this case, the summation was selected because it aggregates the importance of each species linearly, and accounts for their quantities.

The testing of Hypothesis B identified that depending on the combination of CNE, OdC, and DAC, different ranking strategies proved more useful. Figure 57 depicts the distributions of CNE and DAC from the DiMA engagement matrices. The OdC values were not computable because the rank of the matrices was not sufficiently high. Nonetheless, the settings for CNE and DAC can be considered to be low and medium respectively, which in the cases studied previously, would indicate that a ranking based on the absolute FV would be the most suitable. The distributions of absolute FV are presented in Figure 58. In this figure, the distribution of absolute FV is presented for each of the 20 species in a different histogram. The abscissa is the value of absolute FV, and the ordinate is the number of observations for that particular range of values. Species with higher ranking absolute FV values should have a mean greater than zero, an possibly a skewness in the positive direction. In this case, the distributions do not favor any particular species, but seem to indicate that according to the FV they are all similarly critical. Studying the distributions of PFE, presented in Figure 59, leads to a different conclusion. In the case of the PFE there are groups of species that display higher rankings. Specifically, species 20 clearly displays a skewed distribution shifted to the right. Other species display similar

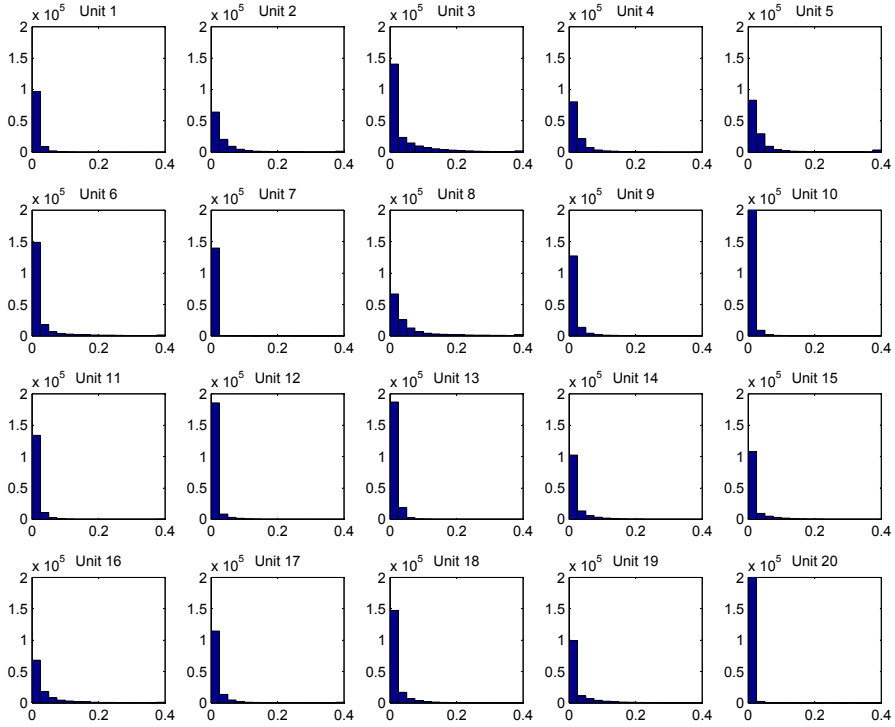

Figure 58: Fiedler Vector for the gDNOsim test of Hypothesis B.

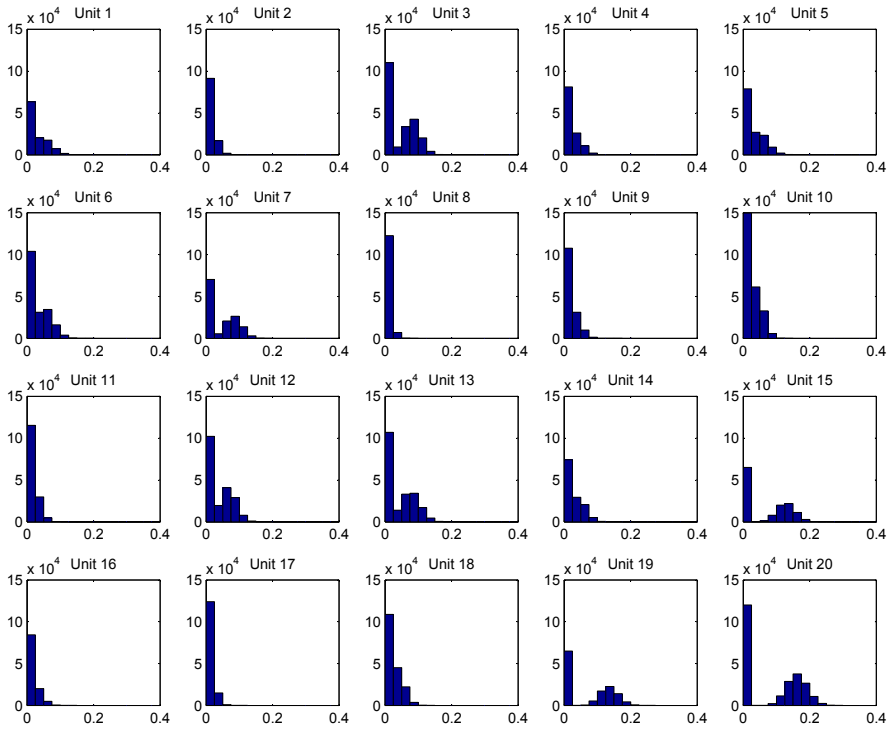

Figure 59: Perron-Frobenius Eigenvector for the gDNOsim test of Hypothesis B.

characteristics, although their degree of skewness is less severe, e.g., 19 and 15, followed by 3, 12, 7, 13, and 6. This indicates that in general terms, the PFE component for these species was higher than for the other species. According to Hypothesis B the cases which had error infused to species 20 should display the largest amount of error when compared to the *reference model*. Similar effects results should be displayed by the other species with skewed distributions, but not to the same degree.

Each of the 20 deviations evolved differently than the *reference model*, this deviation must be studied to determine which evolution was closer to the reference model. For this purpose, the following metrics were used to assess the difference: (1) the final number of units killed per side and (2) the fraction of surviving species. *Models of the model* that are good representations of the *reference model* will have small differences in these two metrics,

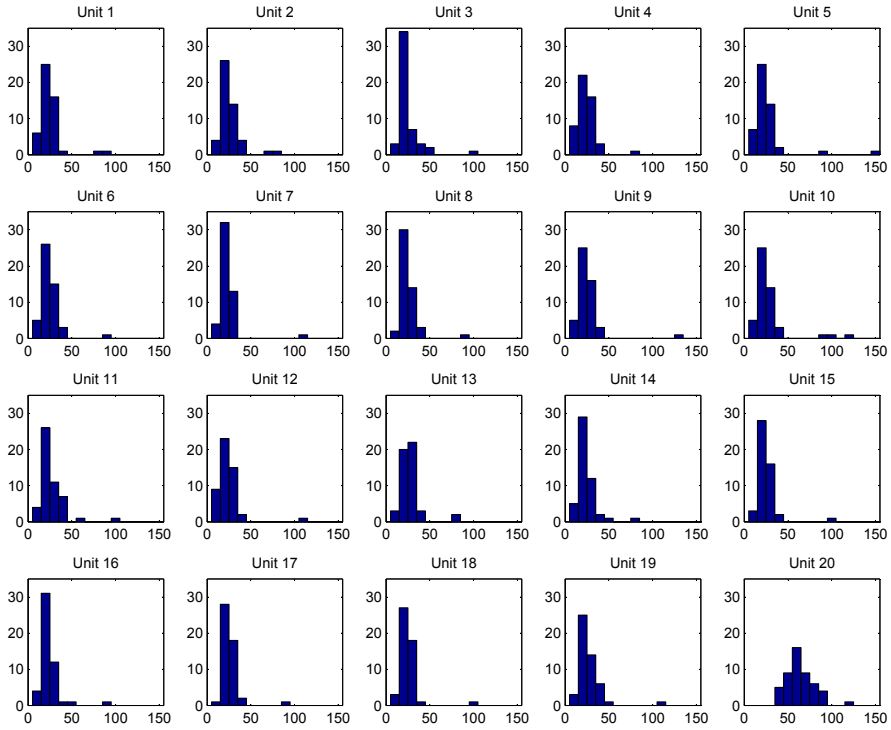

Figure 60: Error in predicting number of casualties per species for the gDNOsim test of Hypothesis B.

whereas models that do not capture the behavior of the *reference model*, will deviate and produce larger variations. The difference in the metrics will be computed as the summation of the absolute value of their difference. According to Hypothesis B, if a species is critical, then the *model of the model* that has the error infused to that species should display the largest difference. In addition, a critical species should have higher values in the FV and/or PFE than a non-critical species. These two rankings were assessed from the engagement matrices created by DiMA.

Figure 60 depicts the error in number of casualties per species for infusion of error into each of the 20 unit types. In the distribution of PFE rankings, it was observed that the last species of agent displayed a tendency to receive a higher ranking. If Hypothesis B is correct, when error is infused on the 20^{th} species, the *model of the model* must display

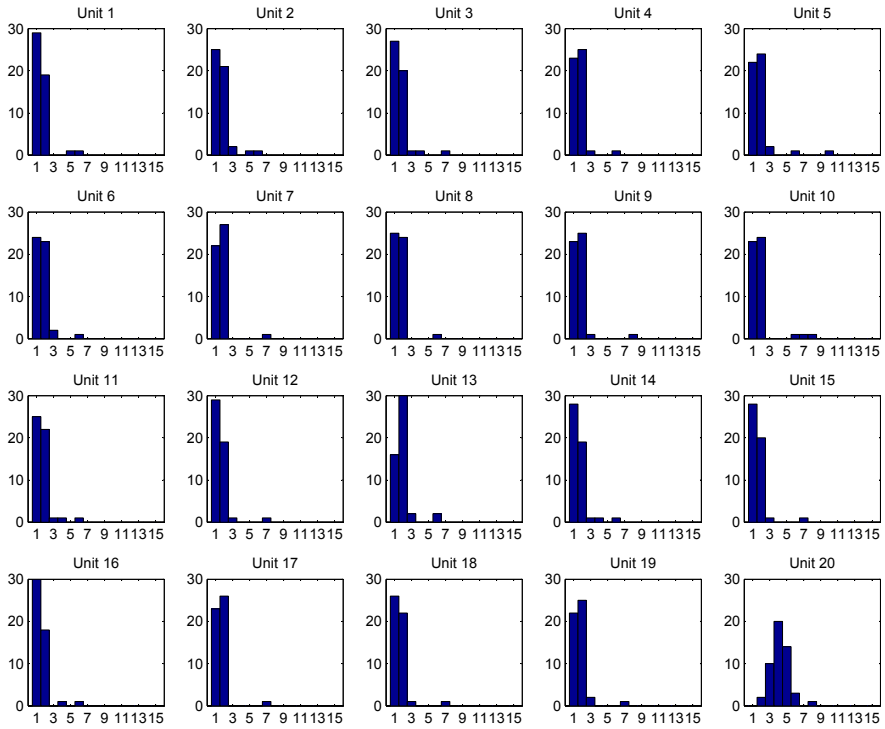

Figure 61: Error in predicting number of casualties per side for the gDNOsim test of Hypothesis B.

the most amount of difference in the number of casualties per species. This is clearly the case depicted in Figure 60, an indication that for this system, the PFE ranking identifies the critical species. The error in the casualties per species is not as distinctly different for when the other species display error. Figure 61 represents the distributions of error for casualties per side. The distributions are similar to the ones presented in Figure 60, once again, indicating that for these two metrics and this model, the PFE ranking was able to identify the critical entities.

In summary, the engagement matrices created by DiMA can be used to identify the critical species in a constructive simulation without resorting to creating the constructive simulation itself. The results presented in this last section are not meant to be thorough tests of the hypothesis, simply to serve as an illustrative example of how the rankings can be used to leverage the constructs already created to analyze large-scale system architectures and use them to guide the creation of more demanding and costly analysis tools.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

At first glance, this dissertation may not seem relevant or even related to aerospace engineering. It is not the intention to make it seem more visionary than necessary, but the author recognized that aerospace designers are becoming more concerned with the larger context–either due to requirements given to designers, or due to a need to integrate more systems into an organic whole. The need to understand the impact that the system of interest will have on the larger SoS is crucial to this undertaking. This thesis attempted to aid designers in two ways, (1) help them obtain insight into the relative goodness of different high-level architecture options in a cost-effective manner, and (2) when more demanding techniques are required, to help analysts identify the key elements of the architecture so that effort can be focused on those. The ideas presented here may be as applicable to aerospace engineering as to computational ecology, biology, chemistry, and sociology. This caveat was not intentional from the beginning of this quest, but a by-product of the nature of the problem. If anything, it is a testament that the tools employed here, are true to the original formulation of complexity science, in that there is a common nature to complexity, regardless of the discipline or field of study.

An honest dissertation will not only highlight the positive and encouraging results, but also the less encouraging ones because to those readers that will embark in similar ventures, the latter are often more valuable than the former. For this reason a conscious effort will be made to summarize the encouraging and not-so-encouraging results and insights of this body of work.

6.1 Resolution of Hypotheses

The title of this section may not be entirely proper since it is arguable whether or not a hypothesis may or may not be resolved. The intent is to test the hypotheses, at least initially, and speculate on their validity. As Albert Einstein said, “no amount of experimentation

can ever prove me right; a single experiment can prove me wrong." This has been the underlying mantra by which the hypothesis testing has been undertaken, with the intent to stress the statements, not simply provide supporting evidence in order to convince the reader of their validity.

The two central hypothesis were formulated to address the need expressed by the two objectives, and successively tested in an attempt to test their validity and identify the range of their applicability.

6.1.1 Hypothesis A

Hypotheses A was concerned with enabling the relative comparison of two architectures by simply studying the spectrum of their functional relationship graph. To test this Hypothesis, the author developed DiMA, a simplified modeling technique for architectures that could create representative functional graphs of large-scale system architectures from which network metrics could be obtained. The first step was to test whether or not the metrics captured by DiMA mapped to the capability of interest. This was done by performing a DiMA analysis in every time step of an ABS. The results concluded that if all the effects that can produce the triggering of a capability cycle are captured in DiMA, then its results will correlate well with the ABM&S.

The next step was to test the correlation of the technique using only time independent measures. Unlike the previous test, where the analysis of the functional graphs was performed in every time step, his second test obtained time-averaged probability of occurrence of a functional interaction, and used that to create statistical simulations to test the validity of the technique. This second implementation is the true DiMA implementation. The results obtained highlighted the fact that there is a significant correlation between the spectral metrics provided by DiMA and the more physical metrics provided by the ABM. To be more specific, the blue force's $\lambda^{\mathcal{A}}_{1,total}$ is well correlated with the number of red casualties, and viceversa. This test was not able to invalidate the hypothesis that there was a relation between the results obtained by DiMA and the more complex model. When all the architectures executed using the agent based simulation were compared DiMA agreed

more than 98% of the time with the ABM results as to which one was more capable. This means that if the inputs to DiMA are accurate, and the goal is to compare two or more architectures in terms of their ability to fulfill a capability need, DiMA produced the same result as the ABM more than 98% of the time.

The key to being able to perform this type of analysis correctly is creating the EGM correctly. Therefore, there is interest in studying how the EGM varies for a given case. Testing the variability of the EGM for the set of cases studied indicated that unique EGMs correspond to unique conditions of the ABM, meaning that for the cases studied, the EGM captures the characteristic behaviors of the ABM. The fact that the EGM did not exhibit excessive variability for a given type of system is encouraging because if for a given setting of the ABM, the EGM varies considerably, it is an indication that the method is not resilient to stochastic effects. More importantly, such a discouraging result would be an indication that determining the EGM is a futile venture. The reader is reminded that in the real implementation of DiMA, the users will not enjoy the benefit of an ABM&S environment–that would defeat the purpose of employing something like DiMA–instead, the EGMs must be elicited from expert opinion.

6.1.2 Hypothesis B

This hypothesis was–in comparison to Hypothesis A–more extensively tested not because it was deemed more critical, but because the testing devised was more demanding and required–at least in terms of mathematics–more elegant techniques in order to be rigorous and useful. This hypothesis is also the one that has the most applicability to the generic problem of complex systems. The hypothesis in itself was decomposed into two sub-hypotheses, (1) the contribution of each entity to the overall system's behavior is correlated to their functional centrality, and (2) focusing modeling effort on those central entities will yield the highest fidelity for the overall system.

The tests devised for these sub-hypotheses were based on a simplification of ABM which is similar to the RBN formulation developed by Kauffman in the 1960s. The simplification used in this thesis is not exactly the RBN formulation developed by Kauffman, but for

sake of succinctness the small differences will be neglected and the models be referred to as RBNs. Using this simplified formulation a large number of cases can be executed and analyzed under a variety of conditions and measures of fidelity. An analogy to the reductionist and holistic modeling approaches was devised and appropriate metrics were developed to measure the fidelity of each of these. The candidate ranking proposed by the hypothesis was compared to other viable candidates, including random and uniform rankings (these were included to test whether intelligent ranking, or ranking respectively were important). In addition, a series of additional topological and complexity metrics were computed from the large-scale network to assess the range of applicability of the hypothesis. The key finding from this study is that the three topology metrics chosen–CNE, OdC, and DAC–determine which rankings will offer the best results. High values of OdC and CNE, indicate that the system will behave chaotically and therefore the modeling effort should not be prioritized but should be evenly distributed amongst the entities composing the large-scale architecture. This means that a user can first obtain the functional relation of the entities that compose the complex system. He or she can then compute the CNE, OdC, and DAC and determine which prioritization of modeling effort will yield the best results. The combination of these three (whether they have high or low values, there are eight combinations total), dictate which node metric produces the best ranking of modeling effort per system. The three alternatives are that either ranking is not beneficial, or that ranking proportionally to the Perron-Frobenius Eigenvector or the absolute value of the Fiedler Vector yields the best results.

Despite the fact that a large number of cases were executed for the purpose of obtaining statistical significance, the large number of possibilities in terms of states, topology and rules that even a seemingly trivial RBNs can have makes any statistical sample be minuscule. An analytical study as to the relationship between centrality and criticality was then conducted, but the results could not be generalized to the complete evolution of the network. Nonetheless, the study provided valuable insight as to how a node's criticality is related to how well it is connected to other well connected nodes. This is in essence what the PFE measures, further indication that the spectral centrality of a node may be related

to that nodes ability to impact the architecture the most.

The final section was concerned with integrating Hypothesis A and B together. Hypothesis A was tested by creating DiMA and testing its ability to capture the relative goodness of any two architectures. In the process, DiMA created functional graphs which can be analyzed in the same fashion that the topology of the RBNs were analyzed in the testing of Hypothesis B. A small test was then conducted, whereby an agent based model in gDNOsim with 20 different types of agents, 318 agents in total, were evolved 50 different times. For each evolution, two other sets of simulations were conducted. The first was a 100 repetitions of the DiMA process using time averaged data from the the ABM model. The engagement matrices from DiMA where then used to obtain the topological and ranking metrics discussed in the testing of Hypothesis B. The secon set of simulations were 20 deviations of the original ABM, whereby in each deviation error was induced into one of the types of agents. This produced 20 additional imperfect evolutions of the ABM. These imperfect evolutions were then compared to the perfect evolution to see how they deviated in their prediction of the engagement. The error of each one of these 20 was then compared to the aggregated FV and PFE for each of the species. If Hypothesis B is correct, species with higher absolute FV and/or PFE should display larger errors when error is infused to them than to lower ranking species. In the simulation, the overall system deviated the most when error was infused on the 20^{th} species. If hypothesis B was to be tested, this species should receive higher rankings by the absolute FV and/or proportional PFE rankings. The FV results were similar for all species, but in terms of PFE, species 20 displayed a considerably higher value. This supports Hypothesis B, in that the PFE ranking is capable of identifying the most critical species. The reader is reminded that even though time-averaged data from the original model is used to create the inputs for DiMA, the results are independently obtained, and DiMA had no information as to the behavior from the imperfect evolutions, simply time-averaged probability of interactions between the 20 species in the "perfect" model.

6.2 Ideas for continuation

The author is the first to recognize that the work presented in this dissertation is by no means the final word on any of these problems. It is the intention of this section to highlight possible avenues of extension of the work presented in order to aid those who will follow to venture in–what seem to the author to be–promising but unknown realms of complex systems modeling. These humble recommendations are meant to highlight possible avenues of *academic research*, not necessarily the development of encapsulated tools or the refinement of a specific technique. Those are left to the discretion of the reader, but the author recognizes that the tools presented are not mature for a commercial application.

6.2.1 Hypothesis A

With regards to the first main hypothesis, DiMA is a simple technique based on spectral graph theory, which in terms of application, is in its relative infancy. There are possibly a large number of improvements that can be undertaken in this area, not only in the analysis (e.g., in the addition of graph metrics and the insight they may provide), but also in the construction of the model, and in particular, in the elicitation of the data required to populate the models. The fact that the method must first rely on experience and expertise is something the author would like to make clear. The DiMA process is not meant to operate as a black box, the users must be well aware of the mechanics and cannot blindly apply the technique. The data obtained from the experts must be studied, the output from DiMA must be vetted against their intuition, unexpected phenomena must be explained, and it is only then that the technique will help users improve their understanding of the problem.

In terms of the specifics of DiMA, questions of how to handle competing and dissimilar capability cycles (in the cases presented here there was only one capability of interest) is very much a matter of debate. The proposed manner of calculating an inert and active network to determine the net capability is also an area that may offer great opportunity for improvement.

It may be possible that DiMA is not the best technique, but a variation on it may

provide the insight required. Users that are well aware of the graph theoretic metrics that can be obtained from these stochastic graphs are more likely to be able to tailor the technique to their applications. The author would like to clarify that the work here is not meant to signify the universal solution to the modeling of large-scale system architectures or systems-of-systems. This is a particular application, created for the purpose of testing the hypothesis that an architecture's capability may be measured, at least in relative terms to other architectures, by studying the functional cyclicity of that architecture. DiMA was meant as a tool to test that hypothesis, DiMA is not the main focus of this dissertation.

6.2.2 Hypothesis B

The second main hypothesis also offers ample opportunities for additional research. The RBN formulation employed relied on static networks, it would be interesting to analyze the impact of dynamically changing functional networks which are more on par with the networks observed in ABM&S. This would require the adaptation of the network over time according to a probability of occurrence of a functional relation (or in the case of an RBN, an edge). In addition to this dynamic network reconfiguration, it will be essential to ensure that the boolean rules of the nodes are properly represented by the dynamically changing network. This will enable testing the effect that dynamic networks have on the criticality of a node's behavior on the overall network's behavior. To ensure validity of the results it will be important to reproduce the dynamic changes of the reference RBN in the model RBNs. The boolean constraint of the RBNs may also be lifted by adding more states to the nodes, this will increase the complexity of the rules unless a different formulation for the rules is employed. For example, instead of using truth tables, arithmetic functions may be employed to determine the state of a node. This will require careful consideration of the functions to be employed and how they relate to ABMs.

As a final recommendation regarding the second hypothesis, an analytical formulation relating behavior to structure is critical. The reason the analytical analysis will be fruitful is that regardless of the fact that the RBN formulation is a simplification of the ABM&S

formulation, the number of possible configurations is astronomical (in fact is beyond astronomical), rendering empirical results a dangerous extrapolation. It may be possible to formulate an analytical relation between structure and behavior through the use of Markov Chains.[1] The analytical study presented at the conclusion of the hypothesis testing chapter was not sufficient to relate the entirety of the network's characteristics to the behavior of each node. The study did highlight the importance of the "non-trivial" in-degree of the nodes, but it failed to relate it to the totality of the network as defined by its adjacency matrix. The author suspects that this "non-trivial" in-degree can be related to the PFE of the nodes since *ad infinitum*, the PFE is related to the rechability of the network.

6.2.3 An integrated process

The main conclusion from the review of modeling techniques applicable to the problems of modeling large-scale system architectures was that of the four techniques most commonly employed, each one had a strength and detriment, and that they may not necessarily be mutually exclusive, but could be thought of as being complimentary. The synthesis of Hypothesis A and B yielded a marriage of network theory and agent-based simulation. There is no reason to exclude system dynamics and discrete event simulation, and there have in fact been ample number of studies where network analysis of system dynamic models has produced insightful conclusions as to their behavior and aided in the creation and expansion of the models themselves–e.g., structural validation [271] and model creation [259]. System dynamics have already been integrated with agent-based models, allowing each framework to model different phenomena in an efficient and expandable framework.[59, 156, 288] Similarly, DEVS has been leveraged with agent-based models to exploit the best qualities of each. Social Network Analysis has seen applications where graph theoretic models are integrated with discrete event simulations to expand on the capabilities and add the ability to study dynamic behaviors explicitly.[336]

The different frameworks have been integrated for one use or another, these particular applications can be seen as pieces of a puzzle, that when put together, could produce

[1] The reader is reminded that this is simply *daring speculation* from the author.

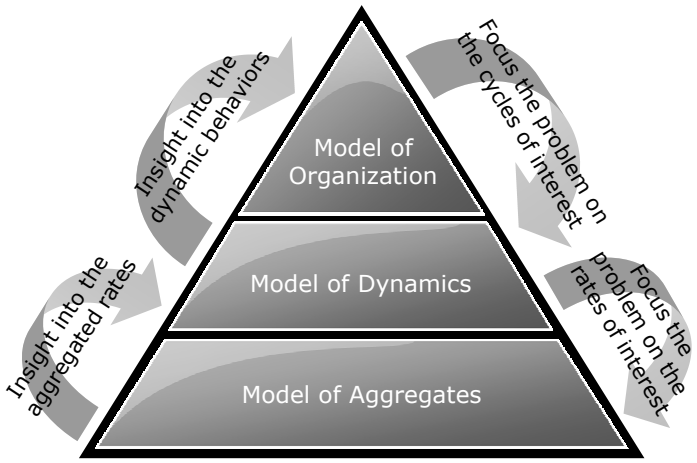

Figure 62: The proposed modeling process in generic form.

an integrated modeling framework for complex systems. Figure 62 depicts the generic form of the process to model complex systems. Each section of the pyramid is a different modeling technique. The idea is to start from the holistic perspective using graphs to capture breadth over depth, and intelligently focus the effort on the portions that drive the behavior, gradually focusing on the critical elements, and then propagate that information up to the higher levels. The model of the organization represents the graph theoretic or network model. The Model of Dynamics represents the SD or the DEVS, where the dynamics can be simulated directly and not inferred from the structure of the functional relations. The SD model would be preferred when the dynamics of the system are better described by continuous relations and the quantities being tracked are so numerous that they can be assumed to be continuous. The DEVS formulation would be preferred when the dynamics of the system occur in discrete steps or events and the quantities being tracked cannot be aggregated into a continuous flow but have to be tracked independently. The modeler can then choose to create one or more ABM(s) to study particular behaviors observed in the other two models, where aggregation (in the case of the Model of Dynamics) or low modeling resolution (in the case of the Model of Organization) obfuscate the analysis.

The insight of the higher fidelity models can then be integrated into the lower fidelity models to improve the results of the broader analyses. The process could be conducted by following steps similar to those presented below.

1. Create an organizational model (digraph) from a list of systems, their compatibilities, and the functions they perform
2. Focus on the portions of the model that provide the critical dynamic behavior by assessing its structure
3. Develop a dynamical model of this portion of the model and determine which aggregates are critical
4. Model the sub-portions that aggregate and compound their effects into the metrics that can be propagated to the dynamic model
5. Propagate the metrics of the dynamic model to the organizational model to analyze the overall capability of the system

Such a process would exploit the most powerful qualities of each type of technique without hybridizing the techniques.

APPENDIX A

TERMS AND DEFINITIONS

A.1 Understanding

Alberts et al. ([20], p. 3) argue that "developing an understanding of how and why things work as they do, or could work, is fundamental to being able to systematically improve functionality. Without such an understanding, progress will continue to be a hit or miss proposition. Understanding enables us to focus attention on making those changes that are most promising." This quote elegantly portrays the importance of understanding, in that it is essential to good decision making. Sun Tzu in The Art of War ([323], pp. 9) states that "If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle." The importance of knowledge and understanding in battle is ever more critical, especially in the information age where the cognitive domain has been recognized as a critical portion of Network Centric Warfare.[78] But what does understanding mean? What is understanding? And how does it related to knowledge?

Einstein said "Any fool can know. The point is to understand." This is a great clue as to what understanding truly means and that it is not knowledge in itself, just because you know does not mean you understand. Understanding, as knowing, is a cognitive concept. The Oxford English Dictionary [6] defines it as: "intelligent, capable of judging with knowledge," "With the: The faculty of comprehending and reasoning; the intellect." Of the verb "Understand" [5] it says: "To comprehend; to apprehend the meaning or import of; to grasp the idea of," or "To comprehend by knowing the meaning of the words employed; to be acquainted with (a language) to this extent," or "To have knowledge or information, to learn, of something."

These definitions are not sufficient to describe what is meant by understanding in the

context of this thesis. As mentioned previously, Alberts et al. ([20], p. 3) related understanding to being able to define "how and why things work." Previously in this thesis, it was stated that understanding was synonymous with being able to answer question of the "why" form. So should it be related to "how" and "why" things work, or just to the "why?" This concept may be easier to grasp if the hierarchy of data, information, knowledge and understanding–often referred to as the Meaning Hierarchy [205], or the Data, Information, Knowledge, and Wisdom (DIKW) Hierarchy [12, 14]–is explained. The following hierarchy is based on Landauer's, who built on the work by Ackoff.[1]

Data Consists of symbols that represent objects, events, and their properties. Raw numbers without context. For example, the speedometer in a car presents data.

Information Information is data that has been made useful. It answers who, what, where, when, and how many questions. It is helpful in deciding what to do, not how to do it. For example, the information that you are driving at 120 MPH will help you decide whether to speed up or slow down, but information wont tell you how to do it

Knowledge Knowledge consists of instructions and know-how. It answers questions of the "how" type and it enables the decision of which course of action to take. For example, your driving knowledge tells you how to control the cars speed by using the gas pedal and the break.

Understanding Understanding consists of explanations. As it was stated previously, understanding answers "why" questions. For example, if the gas pedal in the car suddenly gets stuck and you know why (because the gas linkage may have been twisted) you can evaluate which course of action to take, either put the car in neutral and turn the ignition off, or press it further to see if it becomes unstuck.

'Understanding' refers to insightful knowledge in the form of mental models that are "a relatively enduring and accessible, but limited, internal conceptual representation of

[1]The main difference between their two approaches, is that Ackoff's hierarchy has wisdom as the highest form of cognition, while Landauer has understanding.

an external system (historical, existing or projected) whose structure is analogous to the perceived structure of that system." [111, 112, 287] Understanding therefore, is subjective in nature, since in essence it can be considered to be insightful knowledge. Nonetheless, the test for whether someone truly understands something or simply knows it or about it, can be done by the toddler version of the Socratic method, which in essence will prove that no one truly understands something, since repetitive questions of why something is so, will inevitably leave the person being asked without an answer. The fact that this process cannot be taken to the extreme does not mean that there is no value in understanding why something behaves in the way it does. It is important to understand why things work to know how they will react under different conditions. In a complex system this is particularly difficult since relating causes to effects–i.e., understanding–is clouded by complex causality. Understanding then is analogous to identifying the driving causes to the effects, which in itself is the essence of science, observe a behavior, hypothesize a mechanism or relation (a cause to an effect), and test the hypothesis.

A.2 Architecture

The term architecture has a myriad of definitions. The traditional definitions from dictionaries are either the definition of building architectures, e.g., "the art or practice of designing and building structures and especially habitable ones"[1], "The art or science of building or constructing edifices of any kind for human use"[2], generic definitions related to structure, e.g., "a unifying or coherent form or structure"[1], "construction or structure generally"[2], or the computer science related definitions, e.g., "the manner in which the components of a computer or computer system are organized and integrated"[1], "the conceptual structure and overall logical organization of a computer or computer-based system from the point of view of its use or design; a particular realization of this"[2]. The definitions of building architecture tend to focus on three elements, (1) structure, (2) utility, and (3) beauty. Engineering definitions of architecture, tend to focus on the structure and utility of an architecture, e.g., IEEE 1741-2000 standard defines an architecture to be "the fundamental organization of a system, embodied in its components, their relationships to

each other and the environment, and the principles governing its design and evolution." The missing element is the concept of beauty that is easily applied to artistic interpretations, can still be employed by less subjective disciplines such as engineering by appreciating the *attractiveness* of the integrated product. Griendling proposed a definition of architecture that incorporates this perspective and defined it as: "The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, the principles governing its design and evolution, its purpose, and its attractiveness (e.g. functionality, cost, etc)."[154] This is the definition that is most suitable to the goals of this thesis. Furthermore, it is well aligned with the goal of measuring the *attractiveness* of an architecture and being able to compare different architectures based not only on their structure, but on their ability to fulfill a capability, this makes one architecture effectively more attractive than another. To be able to quantify this otherwise subjective measure, is the central goal of this body of work.

A.2.1 Large-Scale System Architecture

A large-scale system architecture is a particular kind of architecture and is a term devised by the author to signify systems, composed of numerous types of different systems. The term shares many characteristics with what has commonly been referred to as a System-of-Systems, but there seems to be disagreement on the exact meaning of what an SoS is, so the author wanted to use a term that would be unambiguous to the reader.

A.3 Functional Graph

A functional graph for the purposes of this thesis is a layered graph (an edge layered graph to be more specific), where the vertices represent systems, and the edges represent relations between the systems. The relations are grouped by the functions that the set of systems may perform with or to one another, where each layer in the graph is a different function. The graph is directed, in that one system may affect another through a given function but not necessarily the other way around, unless specified by a conjugate edge.

A.4 Functional Cycles

The cycles that exist in the functional graph of the architecture. The cycles are progressions of functional relations between system entities that start and end on the same system. E.g., if the capability is to eliminate an enemy target, the capability must start with an enemy target, relate a series of blue units that collaborate to eliminate the target, and finalize with a unit engaging that same target, effectively completing the cycle of functions through the network of systems.

A.5 Cyclicity

This term refers to the number of cycles that exist in a graph. A graph that has more cycles is said to have *higher cyclicity*. It is correlated to the largest value eigenvalue of the adjacency matrix of the graph, and its normalized version, the Coefficient of Networked Effects.

A.6 Functional Centrality

The concept of functional centrality is one that was developed for this thesis. When systems interact with one another through functions, the functional centrality of a system is analogous to the centrality of a vertex within a graph. That is, the more connected a vertex is to other well connected vertices, similarly, in a system, the idea is that a system that interacts with other systems that in turn interact with a large number of systems, have a higher functional centrality than other systems that are not as well integrated.

In graph theory and network theory there are various measures of centrality. Ranging from degree centrality, to betweenness centrality, closeness centrality, and the one chosen for this thesis, eigenvector centrality (i.e., Perron-Frobenius Eigenvector).

Degree A measure of how many edges are incident to the vertices. For directed graphs there are two variations, the in-degree and out-degree centralities. This the simplest definition of centrality, and it does not distinguish amongst two nodes that may have the same number of edges, but may be connected to other nodes which are not as well connected.

Betweenness This is a measure of how critical a node is to the overall network. It is measured by computing shortest paths for every pair of nodes and assigns higher values to nodes that are part of shortest paths than nodes that are not. These are nodes that if studying a flow network, would most likely receive the most requests.

Closeness Closeness is similar to betweenness in that it uses the concept of the shortest path to assign a node's centrality. Unlike betweenness though, this measure computes the shortest path for every node to every other node and uses the reciprocal of the average of all those shortest paths to determine the closeness for each node. Nodes with high closeness have the ability to reach all nodes in just a few hops, nodes with low betweenness are not as well connected to the network.

Eigenvector The eigenvector centrality assigns high centrality values to nodes that are well connected to other well connected nodes. The Perron-Frobenius Eigenvector is the only one that is assured by the Perron-Frobenius Theorem to be of positive value, and is therefore the one that is used.

APPENDIX B

RANDOM BOOLEAN NETWORKS

Kauffman [187, 188, 189] developed Random Boolean Networks as a simple model that exemplifies the complexity of many biochemical systems. The similarity of RBN to ABM&S are significant. RBNs are composed of nodes that have boolean states (either 1 or 0, on or off, active or inactive), their states depend on the states of adjacent nodes and their future state depends on rules in the form of truth tables. RBNs are evolved over time in discrete time steps. Analogously, in ABM&S, the agents are miopic (interact with a subset of agents, generally a small proportion of the total), they have states that depend on the states of those agents they interact with, and they change their state based on rules, these rules can be interpreted to be the truth tables of RBN. The state of an RBN at a given time t ($\mathbf{X}(t)$) can be specified as the vector of states of its N nodes, i.e., $\mathbf{X}(t) = [x_1(t), x_2(t), ..., x_N(t)]$.

The benefit of RBN over ABM&S is that the characteristics of the RBN can be quantitatively computed, whereas the ABM tends to hanve characteristics that are more subjective. For example, the complexity of the behavior of the RBN can be quantitatively determined by studying its periodicity and information content. RBN generally fall within 3 categories, (1) stable (all the nodes reach a steady state), (2) periodic (the nodes follow a pattern that repeats itself, (3) chaotic (a pattern that never repeats itself)[1]. It is often said that complexity occurs between order and chaos [31, 189, 238], in this case, the longer the pattern, the more complex the system becomes. A depiction of this is offered in Figure 63, where the different regimes of a time-series is presented and the perceived state type (i.e., static, periodic, complex, chaotic) is represented by the different color filled regions. The difficulty in determining where exactly the system is "complex" is evident from the picture,

[1]This is not true of Classical RBNs because by their very nature, they have a finite set of states and every state can only lead to one other state, therefore, eventually, the set of all possible states will be visited and a state will have to be repeated, at which point the RBN completes a state cycle and must therefore reproduce the sequence.

Table 18: Comparison between ABM and RBN.

Property	**ABM**	**RBN**
Entities	Agents	Nodes
- Rules	Various	Binary Truth Table
- States	Various	Boolean
Evolution	Discrete-time	Discrete-time
Overall System Complexity	Subjective	Algorithmic Complexity of TSM
Entity Complexity	Subjective	Entropy of Rule Table

unlike the periodic and static states, the chaotic and complex are difficult to discern by the human mind, i.e., there is not a clear boundary where the system is complex, or chaotic. Determining what is complex, and what is chaotic, has been the goal of much research, e.g., [10, 47, 58, 119, 148, 214, 293, 309], for more information on how to measure complexity, refer to appendix D.

The overall complexity of the RBN can be quantitatively calculated by compressing the output, this is analogous to the Kolmogorov complexity (also known as AIC). A random sequence tends to require the most amount of data to compress it, so the length of the compressed TSM of the network must be normalized to that value. Values close to one indicate chaotic behavior for the network, values close to zero indicate periodic or static behavior, therefore, a complex system should exhibit normalized AIC somewhere in-between these two extremes. At the same time, the complexity of the internal rules can also be determined by looking at their entropy, or at how many "active" rules each node has.

The output of an RBN is often represented as a TSM, an example is depicted in Figure 64, where the nodes are listed in rows (there is one row for each node), and time in columns, advancing from left to right with the initial condition usually represented as the left-most column.

When modeling, the effort of modeling can be determined to be the number of rules (behaviors) modeled. As the number of type of agents[2] increases, their behaviors need to

[2]Agents within the same type may have different describing parameters, but their methods are the same,

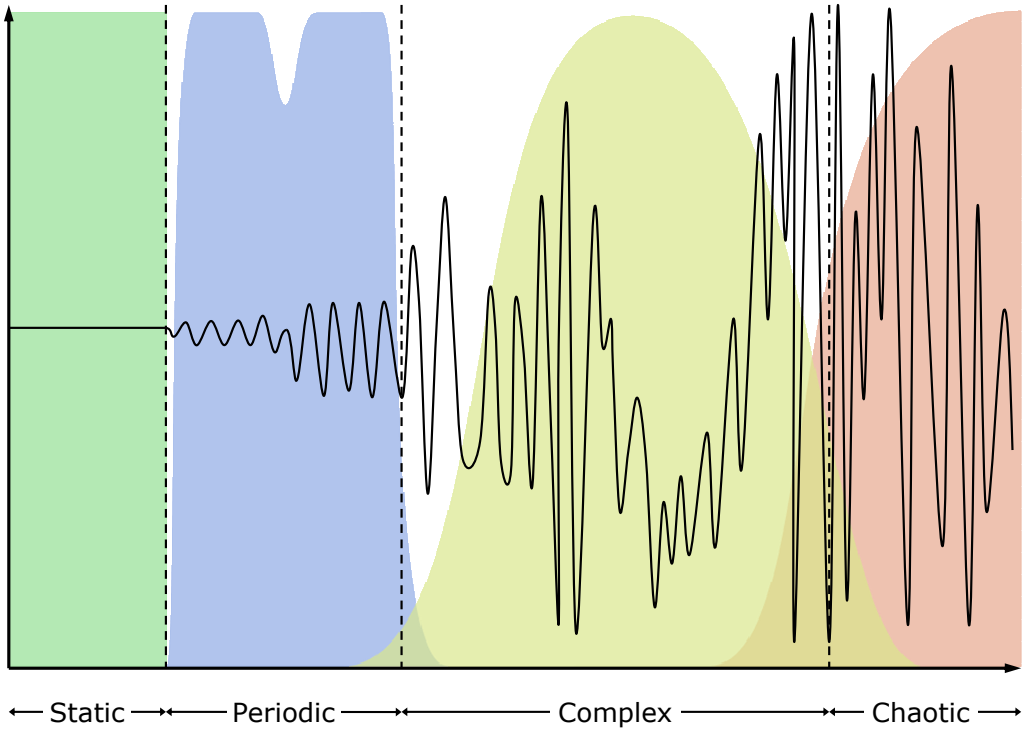

Figure 63: System state regimes.

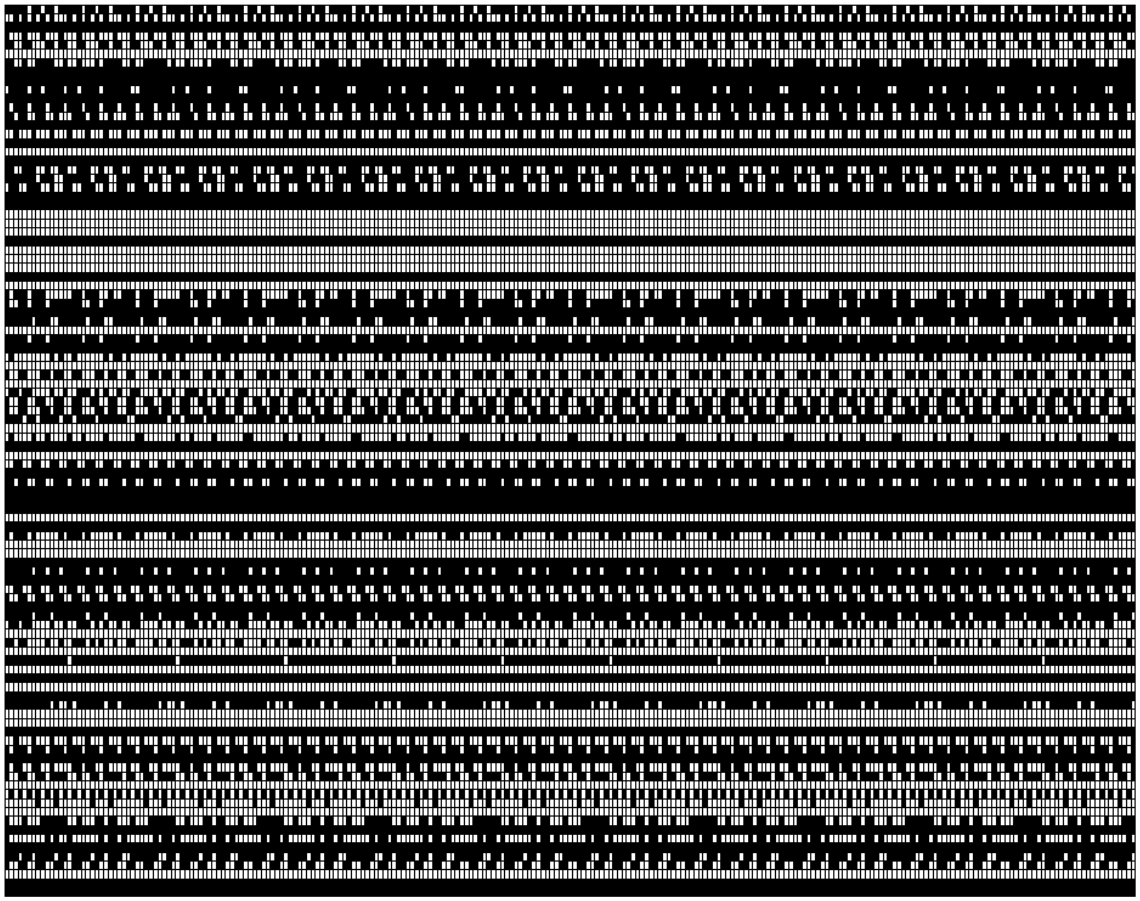

Figure 64: Time-State Evolution of a Random Boolean Network with 100 nodes over 250 discrete time-steps.

be modeled, but since an agent's behavior depends on the state of others, this growth in modeling effort is super-linear. In the case of an RBN, the number of rules for each node is given by $2^{|inputs_i|}$, where $|inputs_i|$ is the number of inputs for node i.

RBN can be evolved deterministically or stochastically. In this work the networks will only be evolved deterministically. The state space of an RBN has rank 2^n, where n is the number of nodes. This superlinear growth with respect to the number of rules, means that the state space of an RBN can quickly become intractable, e.g., the state space of a 100 node network is 1.267×10^{30}. To put context to the size of this number, a stack of that many pennies would span the current estimated size of the universe one and a third (1.33) times![3] Nevertheless, researchers have managed to attempt to understand the state space of these networks. Figure 65 (recreated based on Figure 13 pp. 12 in [352]) describes the evolution possibilities for deterministic RBNs. Since the network evolves deterministically, every state can only lead to one other single state (multiple states may still lead to any one state though). Therefore, if a state is revisited, that means that the network will iterate through a cycle. Figure 65 shows how these trees of network evolution may eventually lead to a cycle (subfigures IV and V). These attractor basins [352] can help organize the 2^n states of the RBN and provide insight into the network's behavior by increasing the traceability of its state space. The difficulty lies in that as the number of nodes increases, the number of possible states increases exponentially, and the likelihood that the network's attractor basins will be characterized by extremely long trees increases, and it becomes impractical to identify the attractor basins explicitly.

e.g., the agent type may be attack aircraft (e.g., they follow the same rules for ingress, egress, carry air-to-ground weapons, etc.), but the specific agents may be F-16s or A-10s, having very different properties (e.g., speed, fuel consumption, RCS, etc.).

[3]The universe is estimated to be 156 billion light-years wide [63] and the average US one cent coin is $1.55mm$ in thickness.

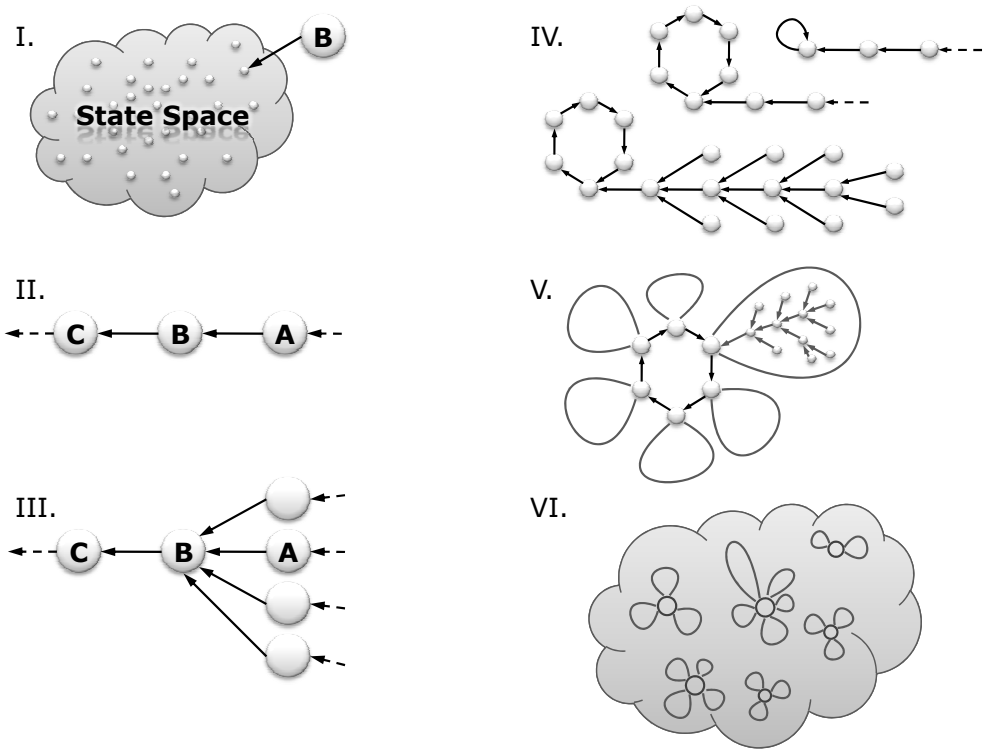

Figure 65: Classifying the State Space of an RBN.[352]

APPENDIX C

LOWER DIMENSIONAL ENCODINGS OF THE TIME-STATE MATRIX

C.1 Decimal Interpretation

In order to obtain well behaved decimal interpretations of the macro state of the RBN_R, two steps are necessary when translating the binary output to decimal encoding. The first is the sorting of the nodes according to their clustering, the Fiedler eigenvector (the first smallest non-trivial eigenvector of the Laplacian matrix) provides a suitable ranking for the nodes, an alternative sorting is provided by the activity of the node. The second step requires that the matrix be decoded using Gray coding to ensure that small changes in the bits do not produce large effects in the decimal encoding. Both of these steps are necessary in order to obtain well behaved decimal representations of the TSM.

Figure 66 provides an example of different decimal encodings of a 100 step TSM. The blue has been sorted using the Fiedler eigenvector but decoded using regular binary encoding. The red has been decoded using Gray coding, but has not been sorted. The magenta has been sorted by the Fiedler eigenvector, and decoded using Gray coding. The unsorted TSM displays a more chaotic pattern than the sorted, indicating grouping according to clustering has a significant effect in the dynamics of the network. The use of Gray coding seems to also improve the behavior of the time series. Results seem to indicate that sorting by clustering (using the Fiedler eigenvector) provides the most impact in regularizing the decimal timeseries.

Alternatively, a structure independent sorting of the nodes can be done by sorting the nodes based on their activity, or average state. Figure 67 shows the TSM of the Rule 110 1-D Cellular Automata unsorted and sorted by its average node state. The patterns are evident in the unsorted case and are more difficult to discern in the sorted. This is an example of why sorting may not always produce the most recognizable results, but at the

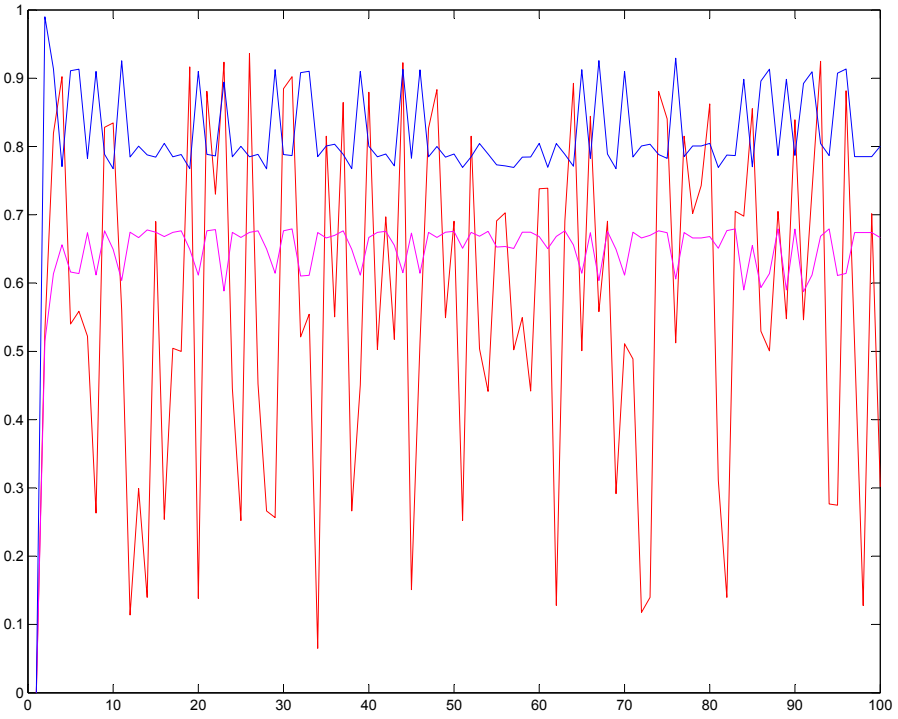

Figure 66: Sorting and decoding of the TSM. Sorting by Fiedler eigenvector with regular binary encoding (blue), unsorted with Gray coding (red), sorting by Fielder eigenvector with Gray coding (magenta).

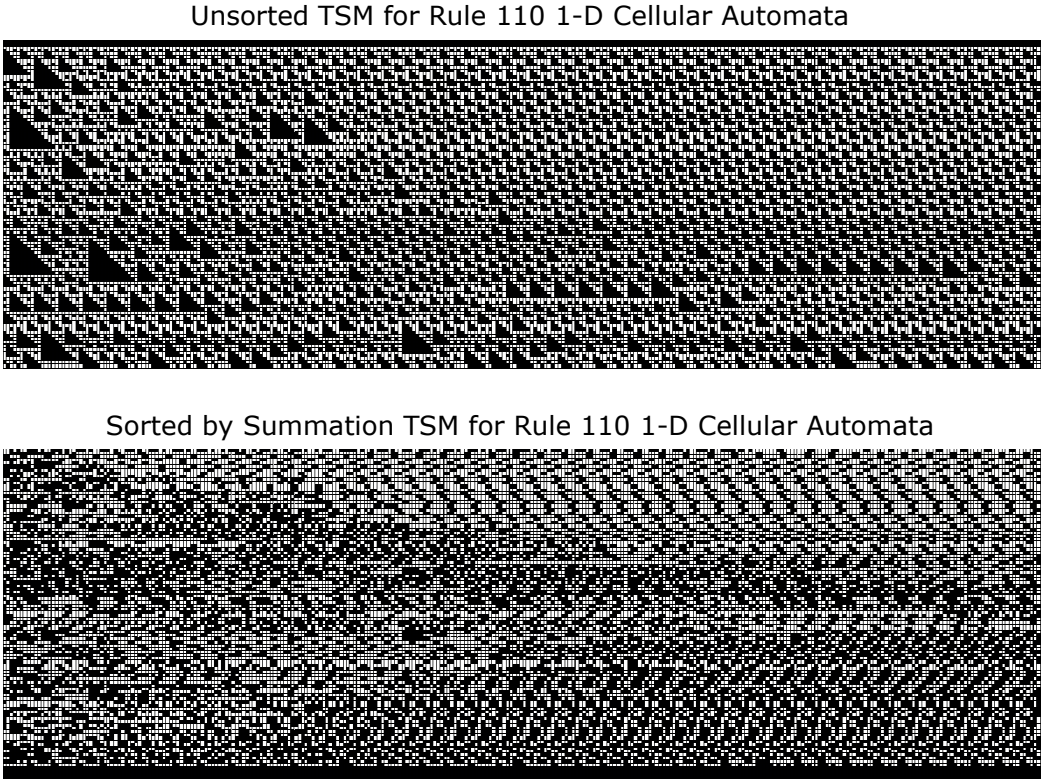

Figure 67: Sorting the TSM based on node activity.

same time, the interest is in supporting automated recognition of these patterns to aid in the classification of the network's complexity.

APPENDIX D

MEASURING COMPLEXITY

There are a large number of research efforts concerned with the measuring of complexity[1], yet all of those can be grouped into two general categories; those concerned with using an adaptation of the concept developed by Kolmogorov, and those using concepts from Information Theory, generally related to entropy.

D.1 Kolmogorov Complexity

The Kolmogorov Complexity – also known as the Kolmogorov-Chaitin Complexity, Descriptive Complexity, Stochastic Complexity, Algorithmic Complexity, or Program-Size Complexity – of an object is defined as the computational resources needed to specify the object. In mathematical terms, the Kolmogorov Complexity ($K(s)$) of an object s is the length of the minimal program that describes it exactly ($|d(s)|$). An alternative definition is the length of a self-halting computer program that can produce the sequence of interest.

This presents a problem, as illustrated by the Mandelbrot set presented in Figure 68. The representation of the Mandelbrot set seems to contain a large amount of information, there are some evident patterns, but how should those be described? And more importantly, how much information is needed to describe them?

It is the surprising case that this is the output of a very simple program. It requires the evaluation of a polynomial, $P_c : z \rightarrow z^2 + c$, where c is a complex number. A point belongs to the Mandelbrot set if its value does not tend to infinity as P_c is iterated starting from the point $z = 0$. This example begs the following questions:

1. How can the minimal program for reproducing a sequence be identified with any certainty?

[1] A web search for "complexity measure" using Google™Scholar yields 13,900 results, complexity+measure yields 2,500,000 results.

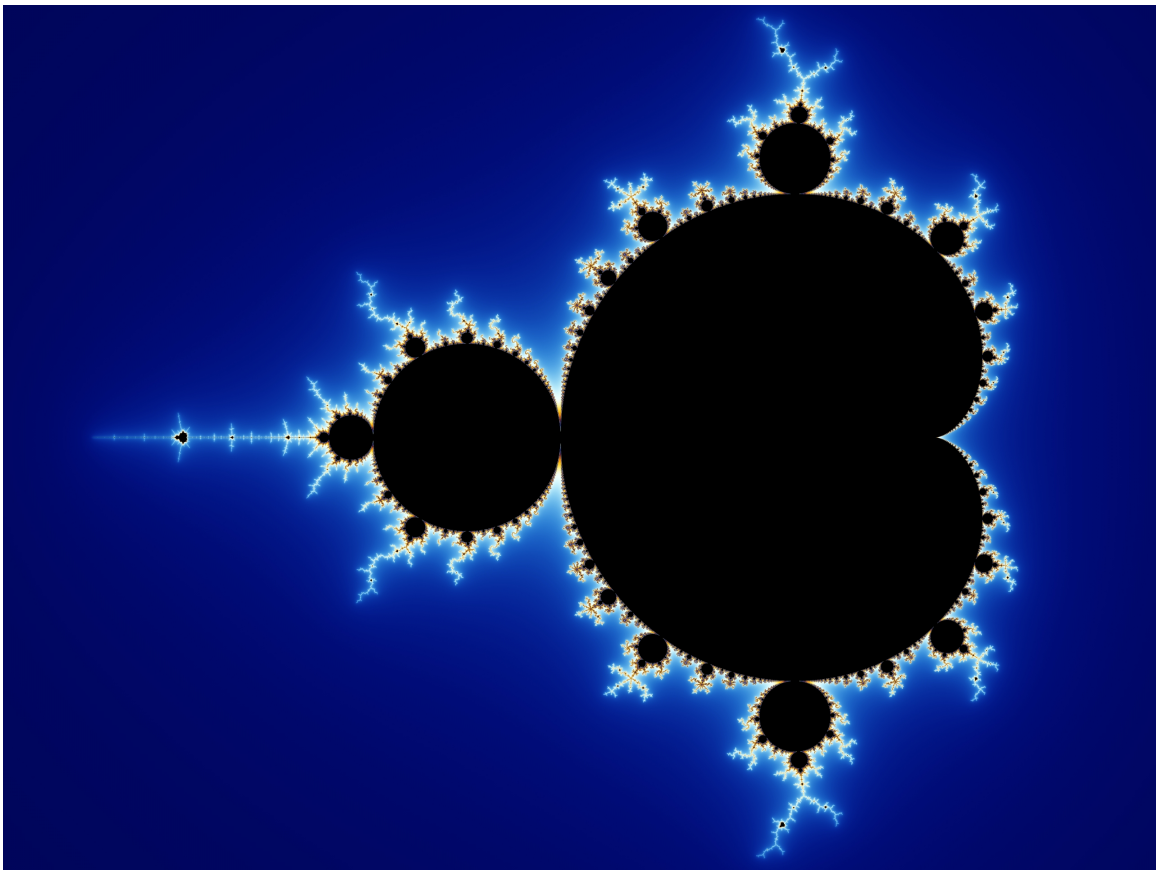

Figure 68: Mandelbrot Set.

2. Serious computational effort is required to compute a Mandelbrot set to the resolution presented in Figure 68. It may be possible to very succinctly describe the process to produce a complicated output, but how much computational effort is required to reach that output?

The truth is that the minimal program to reproduce a complex sequence can never be identified with total certainty. Trivial examples, where the sequence is completely periodic or unchanging can be reproduced with minimal programs, but for complex sequences, such as the Mandelbrot set, it is seldomly possible to ascertain that the program identified to reproduce the sequence is in fact the minimal one. In order to make this a computable measure, assumptions about the types of operations must be made in order to make the number of possible programs enumerable (finite).[2] The question then becomes, is a metric any good if it cannot be measured? The problem with computing the Kolmogorov-Chaitin complexity is in determining the lower bound, not its upper bound. Any compression algorithm can provide the upper bound, the question is how much more can that sequence be compressed. By trying a number of different compression algorithms, it may be possible to ensure that a healthy sample has been studied, but a strict and certain measure of Kolmogorov complexity cannot be found for non-trivial sequences. Solomonoff [307], Martin-Löf [229], and Chaitin [80] proved that the Kolmogorov complexity of a sequence in not computable because of this reason. This is in fact a variant of the Halting Problem, which is related to Gödel's Theorem.

The second question leads to work done by C.H. Bennet [50], where the complexity of a sequence is not the length of the program, but the computational effort required to execute said program and produce the desired sequence. Bennett called his measure "Logical Depth" and defined it to be "the number of steps in the deductive or causal path connecting a thing with its plausible origin."[50]

[2]Lempel and Ziv did this by constraining their program to only two operations.

D.1.1 Lempel-Ziv Compression

As mentioned previously, it is technically impossible to specify the Kolmogorov-Chaitin complexity of a sequence because there is no method to ensure that another program would not reproduce the output using fewer lines of code since there are an infinite number of possible programs.[307, 229, 80] For this reason Lempel and Ziv proposed using only two operations, copy and insert.[209] Their work later became the basis for many of the dictionary coders (most of these coders were named starting with LZ in honor of Lempel and Ziv). As the name implies, dictionary coders record a "dictionary" of terms identified in the data to be compressed. Kaspar and Schuster later developed an efficient algorithm to compute the Lempel-Ziv complexity of a binary string.[186] This algorithm is only meant to compress binary strings, but the output of the RBN has size $n \times t$, where n is the number of nodes and t is the number of evolutions of the RBN.

D.2 Information Theoretic Methods

Information theoretic methods for measuring the complexity of a sequence have focused on entropy related concepts. For example, the "thermodynamic depth" concept developed by Lloyd and Pagels [218] and the concept of statistical complexity by Crutchfield and Young [90]. These measures tend to be theoretically computable, but more demanding in computational effort. For example, Thermodynamic Depth is calculated as the entropy of the set of possible trajectories leading to the current state, which requires computing a large number of trajectories infinitely back in time.

Crutchfield and Young's approach is based on the concept of ϵ-Machines. An ϵ-Machine is a statistical inference procedure used to efficiently model in order to discover the stochastic and deterministic properties of a process. In essence, an ϵ-Machine is a simple computational element that reproduces a given sequence. The structural complexity of the graph of the ϵ-Machine can then be studied. Crutchfield and Young compute the Rényi entropy of the PFE of a probabilistic connection matrix based on the morphs of the trees identified in the string, but other approaches are possible.[85]

D.3 Measuring Complexity in Practice

The benefit from the information theoretic methods is that they tend to maximize complexity not at the chaotic extreme, but at the complex one. Information theoretic approaches recognize the chaotic regime to be as complex as the static regime. The methods based on the Kolmogorov complexity, tend to maximize their value at the chaotic end, which means that they must be mapped in order to measure complexity and not chaos. This concept is illustrated in Figure 69, where the red line represents the value that Kolmogorov complexity would assign locally to that sequence, and the blue line represents the value for an entropy-based or Information Theory-based approach. In theory a mapping between the Kolmogorov-based measurement to the information theory-based can be created since the Kolmogorov-based is monotonic. Since measuring complexity is faster and easier using a Kolmogorov-based approach (e.g., compression algorithm, Lempel-Ziv, Kaspar-Schuster, GNU ZIP, etc.) than using an information theoretic approach, it would be ideal to be able to obtain the type of insight that the information theoretic approaches provide by executing a fast transformation of the Kolmogorov-based one.

D.3.1 Measuring the Complexity of a 2-D Binary Sequence

As exemplified by the extensive body of research on the subject, measuring the complexity of a 1-D binary sequence is not a trivial process, this is only exacerbated for higher dimensional sequences. The output of an RBN is a two dimensional binary sequence.

Most of the systems defined as "complex" are in reality fairly simple in structure (or constitution) but their behavior is complicated. Complicated is an opinion based on the language used, the experience of the observer and his or her culture. Gell-Mann [147, 146] makes the following point. "Imagine an anthropologist approaching a civilization with which he shares a common language, but which is naïve to any culture outside of its own. Now imagine trying to explain to that community a tax-managed mutual fund. What do you think the preamble to that explanation to be?" One thing is for sure, the explanation would be very different than that given to an economics student attending a lecture on fund management. In mathematics language can be an algebra, a series of operations, etc. On the

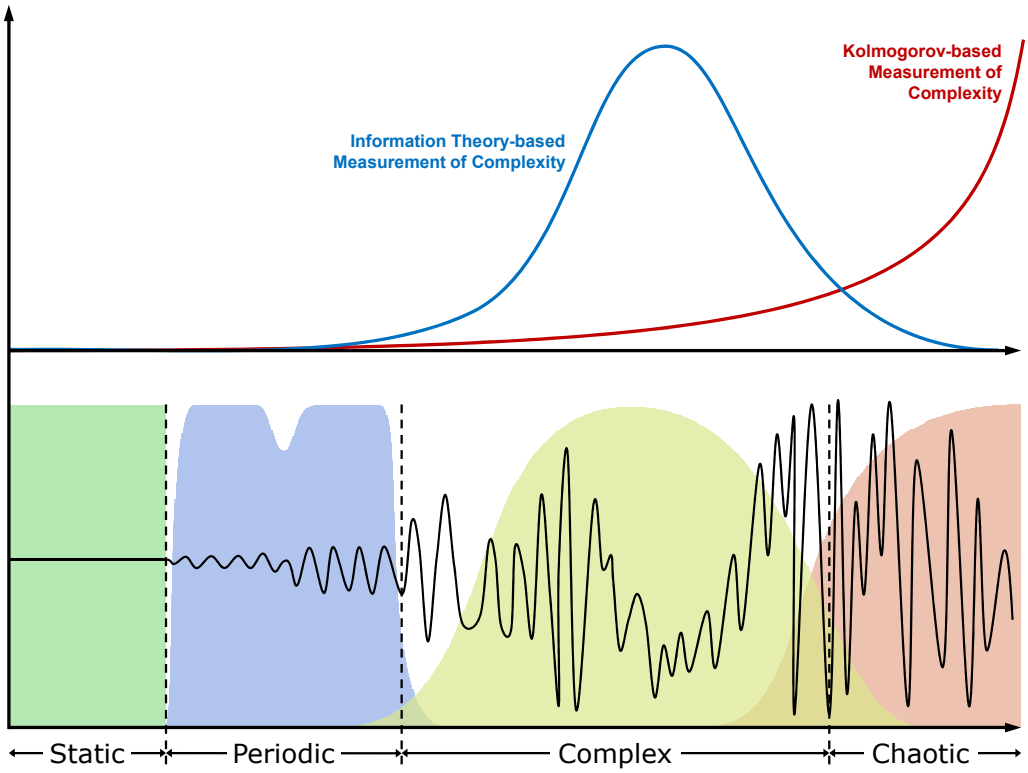

Figure 69: Kolmogorov-based vs. Information Theory-based Measurements of Complexity.

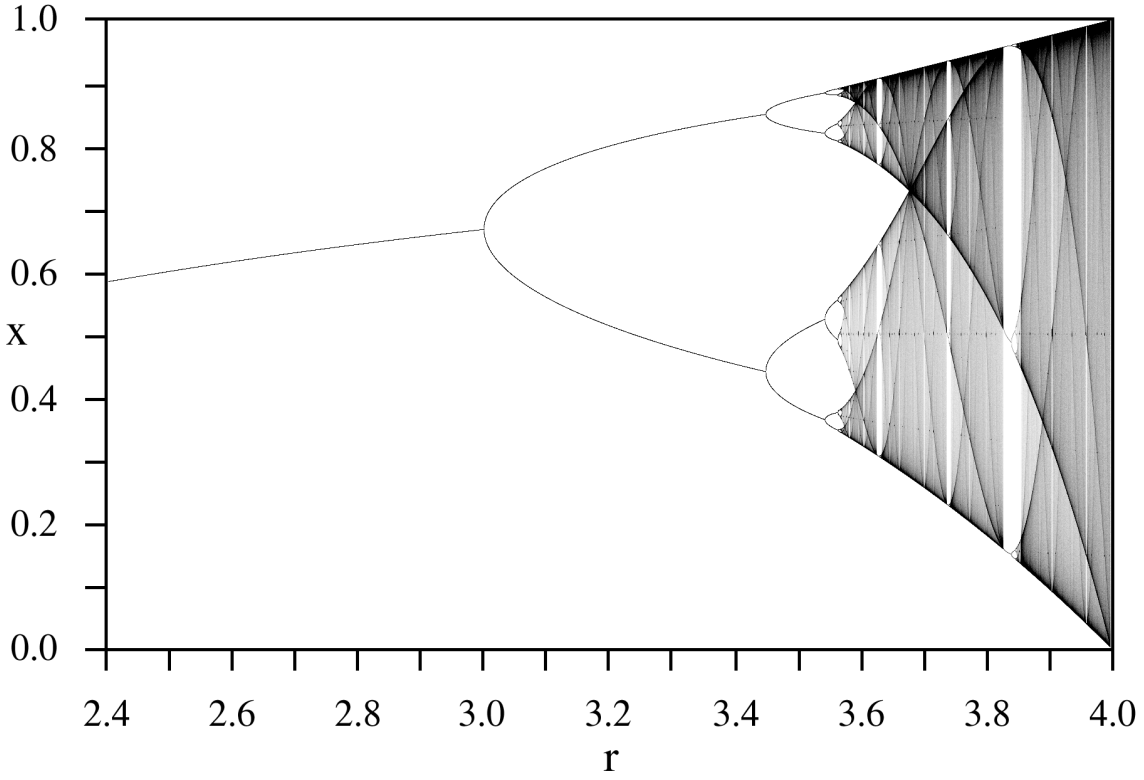

Figure 70: The logistic map bifurcation diagram.

other hand, experience and culture do not translate as easily as language does. If complexity of a behavior can only be defined by the difficulty in explaining it, that complexity means nothing without the specification of a language on which to base the explanation.

Complexity science tends to be interested in two types of systems: (1) systems that are simple to describe but whose behavior is not; and (2) systems that whose behavior is simple but the description of their composition is not. The systems whose composition and behaviors are simple to describe are trivial, and systems that are complicated to describe and whose behavior is complicated as well can only be understood cognitively through abstraction and decomposition, and mathematically with tools that aggregate, as for example those belonging to the fields of probability and statistics.

The most common example of the first type of systems, those that are simple to describe but whose behavior is not, is the logistic map. The logistic map is a non-linear dynamical equation, where the next state, x_{n+1}, is calculated by the function $x_{n+1} = rx_n(1 - x_n)$.

The interesting behavior occurs when r is varied between 3.4 and 4. Figure 70 displays the behavior by showing the number of points the function will oscillate through. As r is increased, the number of points increasing from an organized few, to the chaotic many. The important observation is that the dynamics are created by the same simple non-linear equation, just a simple exponent variation can make the system behave within any of the portions described in Figure 69.

The second type of systems, those whose behavior is simple to describe, but their composition is not, generally imply that there is an organizing principle or mechanism. This is what generally the human mind assumes or at least hypothesizes, since, how can order emerge from chaos?

These two type of systems are often referred to as having "emergent behavior." In the case of the first type, the behavior emerges from simplicity and produces recognizable patterns that are more complex than the entities that created them. In the second case, more order than what was expected is created, therefore, the mind assumes that there must be an organizing mechanism, and in antiquity it was hypothesized that when something like this happened, there was higher intelligence guiding the collective's behavior. We know now with certainty that no such concerted effort is required, simple rules can create order, therefore, we have translated these systems from the second to the first kind, simply by understanding their rules and mechanisms better.

D.4 Measuring the Complexity of a Graph

As measuring the complexity of sequences has been the focus of much research, more recently, measuring the complexity of relationships has been of considerable interest.[236, 85] This has been traditionally done using the *graph thickness* and *coloring number*, but these tend to have low resolutions.[85] More advanced approaches tend to be computationally expensive, e.g., *Medium articulation* for weighted graphs [341] and a concept based on the network motifs [236]. An additional approach by Machta and Machta [223], calls for characterizing the complexity of a graph by the *computational depth of an ensemble of graphs*. This metric is nontrivial to compute and does not assign a single complexity value

to each graph.[85] A more recent approach by Claussen [85] calls for measuring complexity by studying the correlation between nodes, in an approach that may be more analogous to the entropy approaches for measuring the complexity of binary sequences. The algorithm presented by Claussen has been implemented in Matlab and is presented in section F.2.3.

Figure 71 represents the OdC of three different types of random graphs. The Erdos-Renyi G(n,p) model, the Watts-Strogatz Small-World model, and the Barabási-Albert Scale-Free model. Note that the Watts-Strogatz has in general the least complexity, because their model first creates a lattice (very ordered graph) and perturbs it by a small amount (3% of the edges in this case). The G(n,p) model is completely random, but the complexity of these graphs is less than that of the Scale-Free networks created using the B-A model which randomly attaches new nodes with a bias for well-connected nodes. The SFNs are therefore not too random nor too ordered, and therefore, tend to obtain the higher OdC.

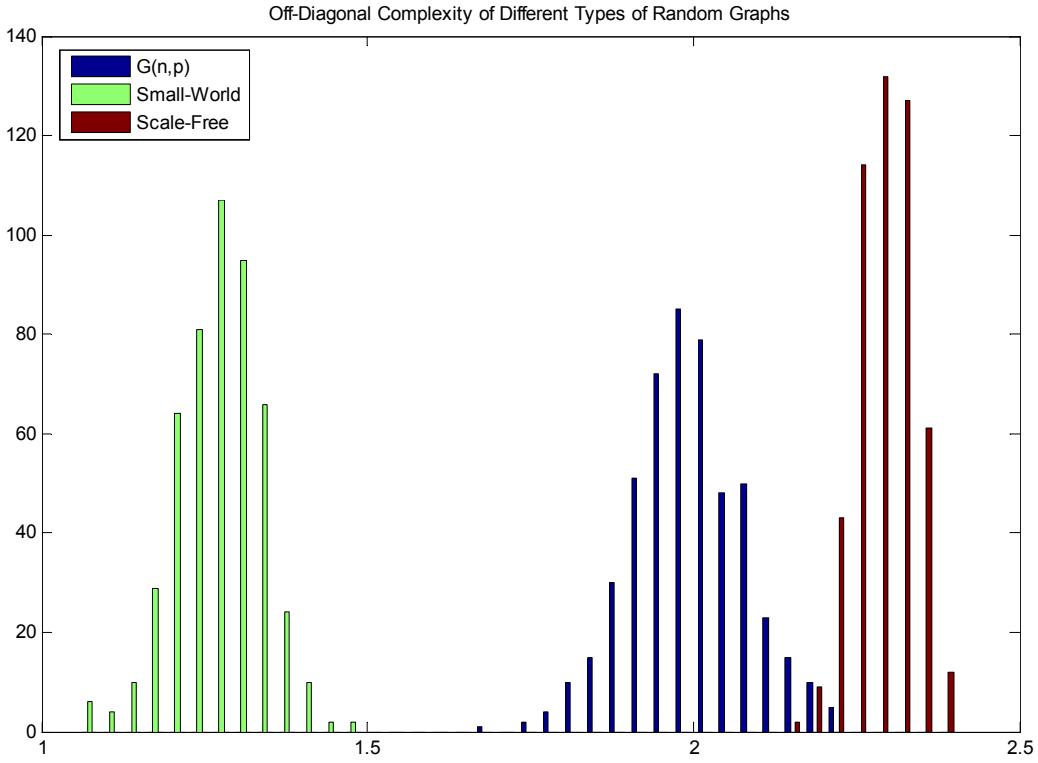

Figure 71: The off-diagonal complexity of three different types of random graphs.

APPENDIX E

NETWORK ANALYSIS USING SPECTRAL GRAPH THEORY

> "I will never forget my amazement at learning that combinatorial properties of graphs could be revealed by an examination of the eigenvalues and eigenvectors of their associated matrices. I now blame my amazement on a lack of education in physics; I'm now shocked when any important property of a graph is not revealed by its eigenvalues and eigenvectors."
>
> - Daniel A. Spielman ([308], pp. 29)

This chapter will study some properties of network analysis that are particularly pertinent to this body of work. It will describe how some spectral characteristics of graphs (those related to the eigenvalues and eigenvectors of the graph) can be related to specific topological characteristics. Extending Prof. Spielman's quote, the spectrum and in particular, the principal component, of a matrix more-often-than-not seems to yield insightful information about the behavior of the system described by the matrix. The reader is reminded that the principal component is the largest eigenvalue and its associated eigenvector. In graph theory, the principal component of the adjacency matrix of a graph gives an indication to the cyclicity of the graph and the 'centrality' of its nodes. This section will break down why this is the case in an attempt to shed light into the meaning of the principal component of a graph.

E.1 Self-multiplication of the Adjacency Matrix

The adjacency matrix $\mathcal{A}_{i,j}$ contains the paths between nodes i and j of length 1. $\mathcal{A}^t_{i,j}$ contains the paths of length t between nodes i and j.

Figure 72 contains a visual example of a graph with its associated adjacency matrix and how they very with powers of t of the adjacency matrix. The adjacency matrix must be read right-up, meaning that it depicts the connectivity of the vertically aligned nodes to

the horizontally aligned nodes. To illustrate this with an example, consider the first node, it is connected to nodes 2 and 3. In the adjacency matrix, entries $\mathcal{A}_{1,2}$ and $\mathcal{A}_{1,3}$ have a value of 1 while the rest contain a value of zero. Note that it is a directed graph and therefore the matrix is not symmetric. The $t = 1$ is the original graph, it contains two cycles, one of length 5 that traverses all nodes, and one of length 4 that traverses all nodes with the exception of node 2. Note that the first self-loops appear with the 4th power, meaning that those nodes can reach themselves after they traverse four edges. On the 5th power of the adjacency matrix, all nodes can now reach themselves as depicted by the populated main diagonal, previously, only node 2 lacked a self-loop because it was the only node missing from the cycle of length 4. On the final power shown, $t = 6$, there is one entry, $v_1 \rightarrow v_3$, which is greater than 1. This means that if six steps are taken, there are two different ways for node 1 to reach node 3, namely traversing the cycle-5 plus the connection between $v_1 \rightarrow v_3$, or traversing the cycle-4 and then $v_1 \rightarrow v_2 \rightarrow v_3$.

E.2 The relation to the principal component

Given a 3×3 matrix $\mathbf{A}$ with eigenvectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ and corresponding eigenvalues λ_1, λ_2, and λ_3, then an arbitrary vector $\mathbf{y}$ can be written

$$\mathbf{y} = b_1\mathbf{x}_1 + b_2\mathbf{x}_2 + b_3\mathbf{x}_3 \tag{30}$$

Applying the matrix $\mathbf{A}$

$$\mathbf{A}\mathbf{y} = b_1\mathbf{A}\mathbf{x}_1 + b_2\mathbf{A}\mathbf{x}_2 + b_3\mathbf{A}\mathbf{x}_3 = \lambda_1\left(b_1\mathbf{x}_1 + \frac{\lambda_2}{\lambda_1}b_2\mathbf{A}\mathbf{x}_2 + \frac{\lambda_3}{\lambda_1}b_3\mathbf{A}\mathbf{x}_3\right) \tag{31}$$

so

$$\mathbf{A}^n\mathbf{y} = \lambda_1^n\left(b_1\mathbf{x}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^n b_2\mathbf{A}\mathbf{x}_2 + \left(\frac{\lambda_3}{\lambda_1}\right)^n b_3\mathbf{A}\mathbf{x}_3\right) \tag{32}$$

if $\lambda_1 > \lambda_2$, λ_3, and $b_1 \neq 0$, it therefore follows that

$$\lim_{n\rightarrow\infty} \mathbf{A}^n\mathbf{y} = \lambda_1^n b_1\mathbf{x}_1 \tag{33}$$

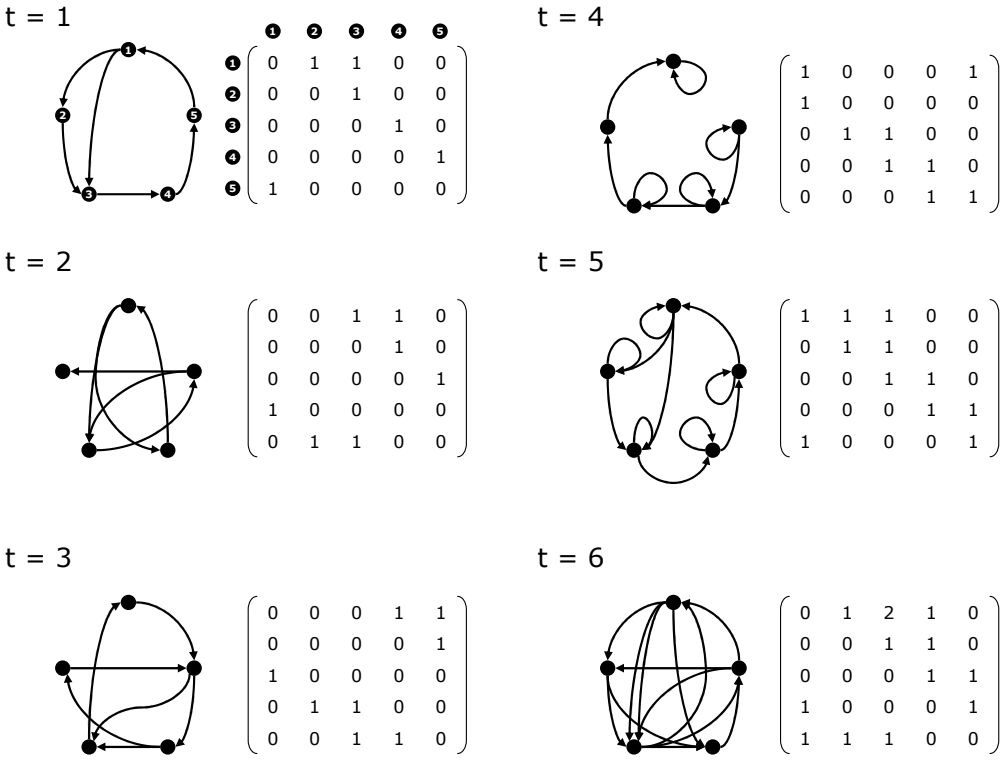

Figure 72: Graph path length and the powers of the adjacency matrix.

so repeated application of the matrix to an arbitrary vector results in a vector proportional to the eigenvector with largest eigenvalue.[1]

As mentioned previously, Jain and Krishna [175, 176, 178] made use of the structure of the graph to understand the dynamic behavior of the system that it described. Their model contains two different rates of simulation, a continuous simulation of the rates of change of the species, and a discrete disruption in the type of the species. In summary, their model assumes that the fast dynamics, the continuous simulation, reaches steady state in-between the discrete disruptions. This is a sensible assumption for the case of fast acting catalytic chemical reactions, or evolutionary changes, but is an assumption that should be carefully considered when studying other complex systems. In their model, the fast dynamics of the system are described by the following equation.

$$\dot{y}_i = \sum_{j=1}^{s} c_{ij} y_j - \phi y_i \tag{34}$$

Where s is the size of the population (the number of species), y_i is the amount of species i, $\dot{y}_i$ is the rate of change of species i, c_{ij} is 1 if species j catalyzes species i, and zero otherwise, and ϕ is the non-equilibrium dilution flux. In their model, Jain and Krishna then define x_i to be the relative quantity of species i, and can therefore re-define equation 34 as equation 35.[175]

$$\dot{x}_i = \sum_{j=1}^{s} c_{ij} x_j - x_i \sum_{k=1}^{s} \sum_{j=1}^{s} c_{kj} x_j \tag{35}$$

In their model, they argue that since x is no longer dependent on ϕ, they can make ϕ zero and not lose generality in the study of the attractors of the model. Attractors are states to which the system tends in steady state. Making ϕ zero, signifies that equation 34 has a general solution, $\mathbf{y}(t) = e^{Ct}\mathbf{y}(0)$, where $\mathbf{y}$ is the s dimensional column vector of populations and C is the matrix containing the catalytic relations c_{ij} ([175], pp. 5685). Their argument proceeds as follows: if $\mathbf{y}^\lambda \equiv (y_1^\lambda, \ldots, y_s^\lambda)$ viewed as a column vector is a right eigenvector of C with eigenvalue λ, then $\mathbf{x}^\lambda \equiv \mathbf{y}^\lambda / \sum_i^s y^{\lambda_i}$ is a fixed point of equation

[1] Weisstein, Eric W. "Eigenvector." From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/Eigenvector.html

35. Then if λ_1 is denoted as the eigenvalue of C which has the largest real part; $\mathbf{x}^{\lambda_1}$ is an attractor of equation 35. The theorem of Perron-Frobenius for non-negative and irreducible matrices, states that λ_1 is real and ≥ 0 and there exists an eigenvector $\mathbf{x}^{\lambda_1}$ with $x_i \geq 0$. If λ_1 is non-degenerate, then $\mathbf{x}^{\lambda_1}$ is the unique asymptotically stable attractor of equation 35, $\mathbf{x}^{\lambda_1} = (X_1, \ldots, X_s) = \mathbf{x}(t_\infty)$. Since Jain and Krishna assumed that their continuous dynamics were much faster than their discrete disruptions, this allowed them to use this attractor to compute the steady state value of their continuous system without performing a numerical integration. It is important to stress once again that in order for this assumption to be valid, the continuous system must be considerably faster than the discrete disruptions.

Since this model is the seminal paper for much of the literature related to the significance of the PFE, the Jain-Krishna model has been implemented in MATLAB.[2] Figure 73 contains examples of the output of the model, which correlate satisfactorily with the results presented by Jain and Krishna in their publications.[175, 176, 177] The traditional model was extended to enable the use of scale-free networks as the initial network and to enable preferential attachment of the new species introduced in the discrete disruption to the most populated species. The effect of preferential attachment can be clearly seen from the two example charts. In these charts, Jain and Krishna's original nomenclature is maintained and the models are executed for 7000 iterations. All four runs are for an m value of 0.25, and a total number of species, s, of 100. The solid line represents the total number of active species, s_1. The stability of the autocatalytic sets seems to be more dependent on the preferential attachment of the new species than on the topology of the initial network.

$$\dot{x}_i = F(x_i) - \sigma \sum_{j=1}^{N} c_{ij} H(x_j) \tag{36}$$

$$x_i(t+1) = f(x_i(t)) + \sigma \left[\frac{1}{\sigma_i} \sum_{j}^{N} c_{ij} \left(f(x_j(t)) - f(x_i(t)) \right) \right] \tag{37}$$

Similar studies have used dynamic models of the form presented in Equation 36, where x_i are the dynamical variables, F is an evolution function, H is a coupling function, and

[2] A copy of the code is presented in Appendix F.1.

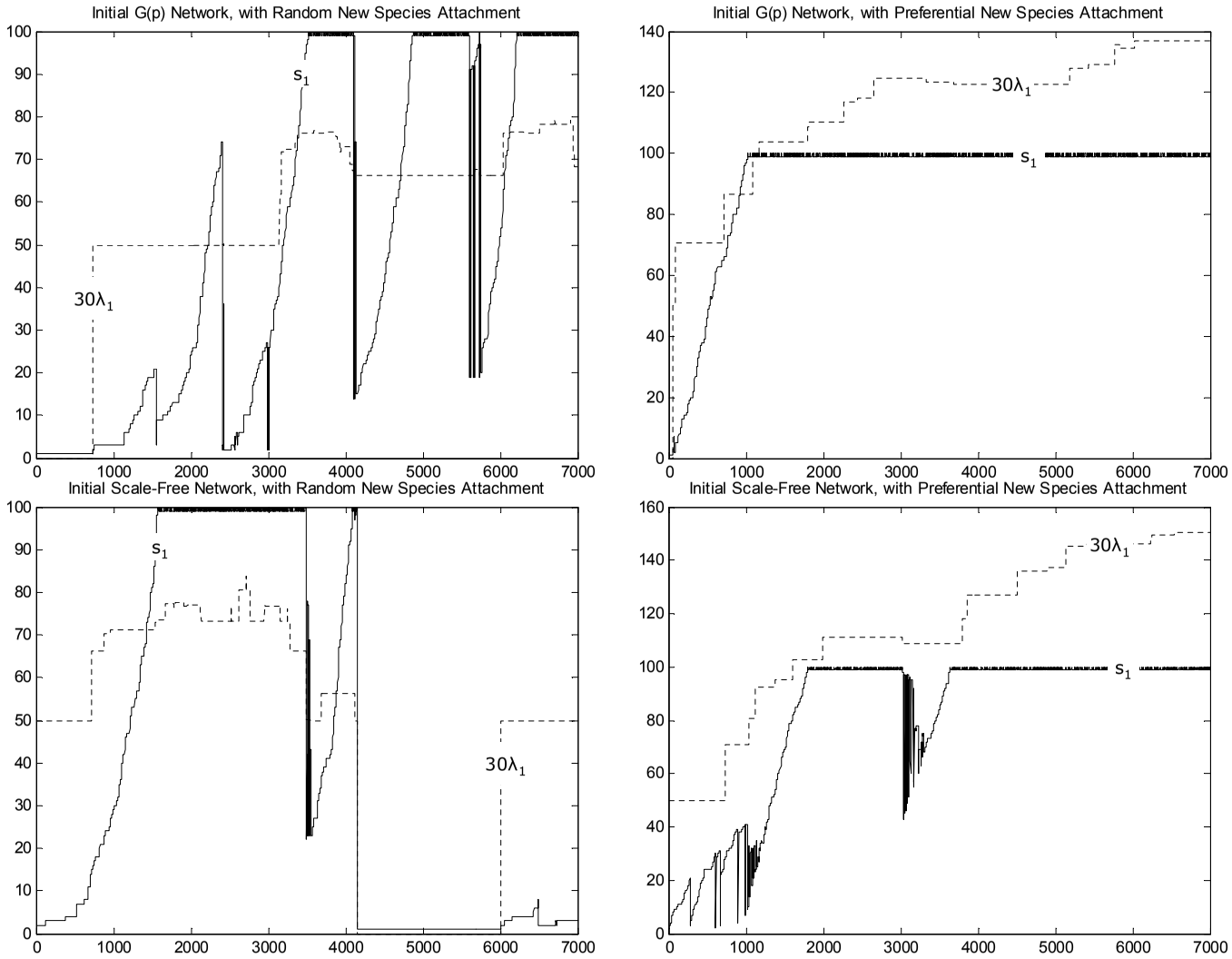

Figure 73: Results from the Jain-Krishna model [175] for random ($G(n,p)$) and Scale-Free Networks.

σ is a coupling constant.[24, 46, 246, 255] Or discrete states as presented in Equation 37.[191] These models have been studied for their ability to synchronize, i.e., reach a steady state and the influence that another spectral parameters has on determining this ability to synchronize. In this case, the spectrum of the representation of the graph is not the adjacency matrix $\mathcal{A}$, but the Laplacian $\mathcal{L}$. The Laplacian is most commonly defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$[24, 191], where $\mathcal{D}$ is the degree matrix, a diagonal matrix with the degree of node i on the $\mathcal{D}_{i,i}$ position.[3] Other representations of the Laplacian include the normalized Laplacian [123], and a weighted Laplacian [263]. One of the most important indications given by the Laplacian is that its smallest nontrivial eigenvalue (λ_2) gives an indication of the ability of the network to "synchronize." Additionally, the number of zero eigenvalues of the Laplacian of undirected graphs indicate the number of components the graph has, i.e., if it has two or more zero eigenvalues, it is said that the graph is disconnected. From now on, the eigenvalues and eigenvectors for the Laplacian will be denoted with a superscript $\mathcal{L}$ (e.g., $\lambda_2^{\mathcal{L}}$, $\mathbf{x}^{\lambda_2^{\mathcal{L}}}$), and those for the adjacency matrix with a superscript $\mathcal{A}$ (e.g., $\lambda_1^{\mathcal{A}}$, $\mathbf{x}^{\lambda_1^{\mathcal{A}}}$), powers of these values will be denoted outside parentheses (e.g., $(\lambda_1^{\mathcal{A}})^2$). As mentioned previously, there have been a myriad of different definitions of the Laplacian matrix, and each definition produces different spectrums. Chow and Kokotovic, and later Fax and Murray, [84, 122, 123] proposed a normalized version of the Laplacian, whereby $\mathcal{L}$ is defined as $\mathcal{L} = I - \mathcal{D}^{-1}\mathcal{A}$, where $\mathcal{D}$ is a diagonal matrix with the in-degrees of each vertex. This version of the Laplacian has also been called the Combinatorial Laplacian, and the $\mathcal{D}^{-1}\mathcal{A}$ been termed as $Markov(\mathcal{G})$ because it denotes the transition probabilities for a simple random walk.[108] In addition, Olfati-Saber and Murray [258] define the Laplacian for a digraph $\mathcal{G}$ as $\mathcal{L}(\mathcal{G}) = \Delta - \mathcal{A}$, where Δ is a diagonal matrix of the out-degree of each node and $\mathcal{A}$ is the adjacency matrix in positive notation, i.e., an edge from node i to j is contained in the (i, j) element of $\mathcal{A}$. The comparison between these different techniques is represented in Figure 74.

The large number of different notations for the Laplacian, and the two different ways

[3]For directed graphs it is important to define whether one is describing the in-degree, out-degree, or a combination of the two.

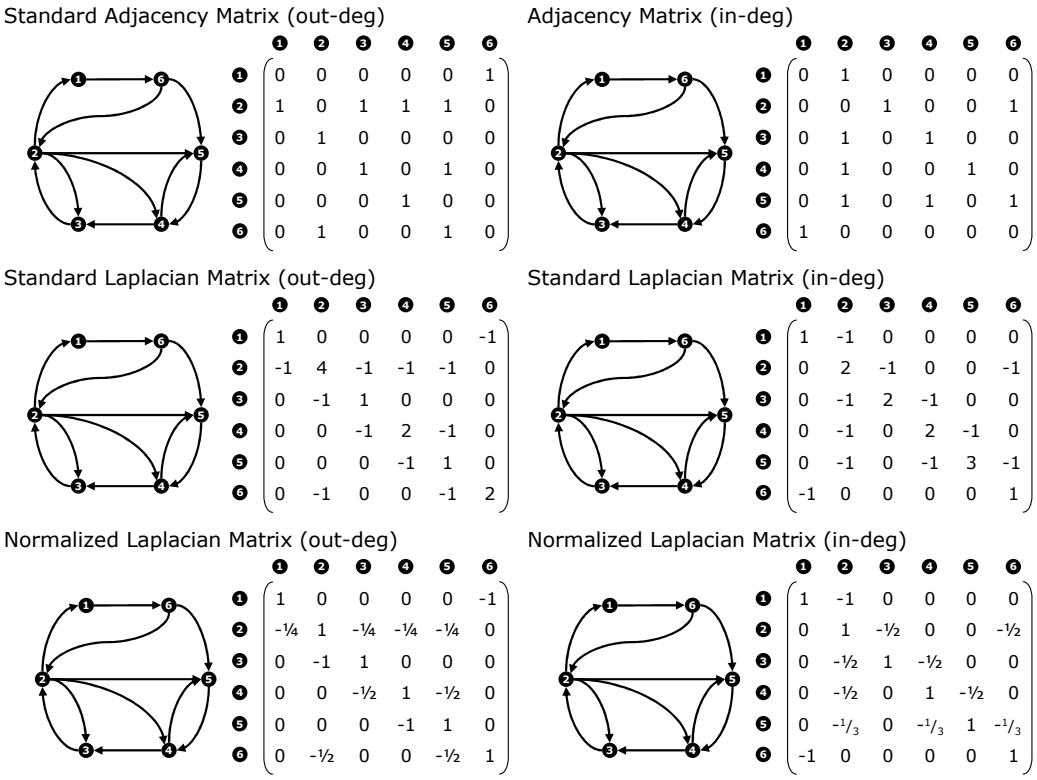

Figure 74: Comparisons between the different definitions of the Adjacency Matrix and Laplacian Matrix for digraphs.

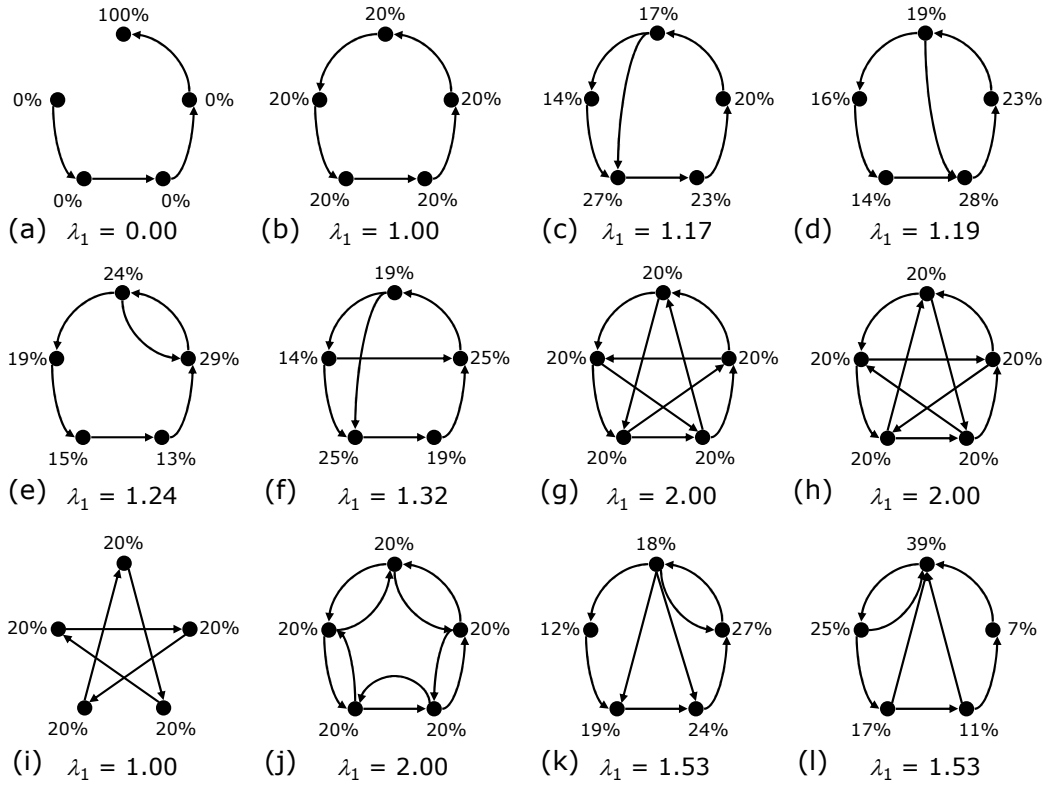

Figure 75: Adjacency matrix's principal component ($\lambda_1^{\mathcal{A}}$) and its associated eigenvector $\mathbf{x}^{\lambda_1^{\mathcal{A}}}$.

of defining the Adjacency Matrix can often be a source of confusion. Most papers in the literature are concerned with the study of undirected graphs, but for digraphs, this is a matter of importance. The traditional negative notation of defining an edge in the adjacency matrix–i.e., an edge between node i and node j is contained in the (j, i) element of the adjacency matrix–may not be intuitive to the novice. Nonetheless, this is the more general approach, and will therefore be the one implemented for every application shown in this thesis unless otherwise noted.

Jain and Krishna's use of the PFE is useful when studying the dynamic behavior of networks that interact through exponential decays and growths, as is the case of the catalyst and evolutionary systems. In this thesis, the goal is more closely related to measuring the cycles and which entities partake in said cycles. The PFE will not be useful in measuring

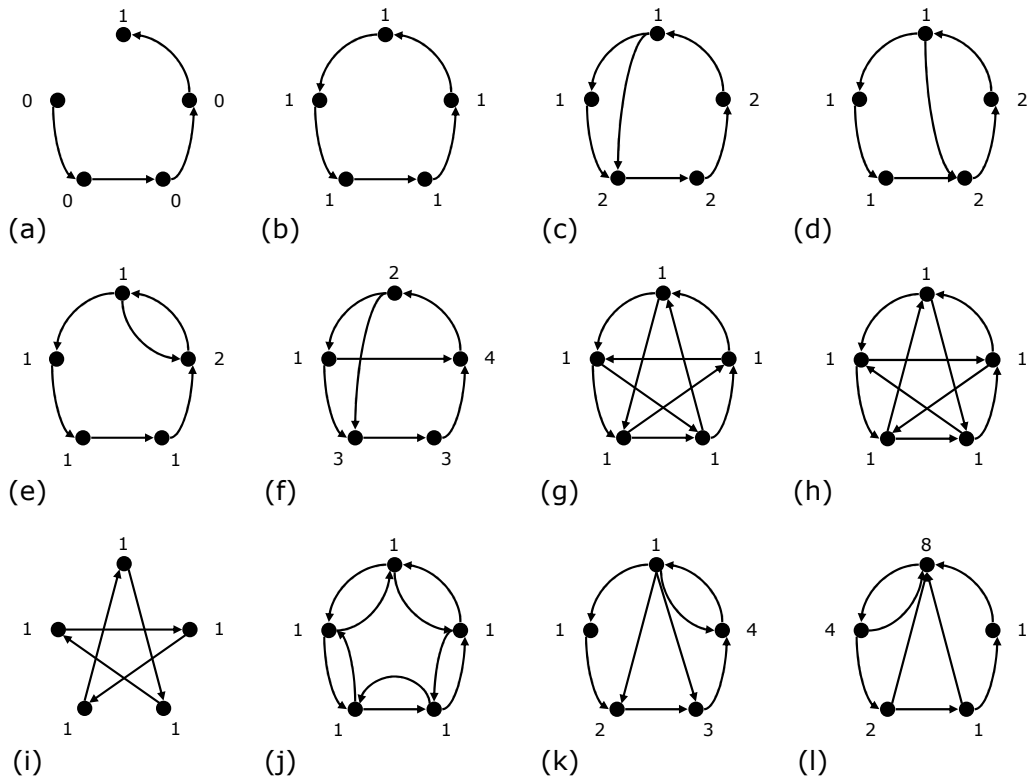

Figure 76: Laplacian matrix's smallest nontrivial eigenvector $\mathbf{x}^{\lambda_2^{\mathcal{L}}}$.

the cycles of a digraph, therefore, a different metric will be employed.

E.3 Study of the PFE and FV

The value assigned to each node by each of the eigenvectors of interest (PFE $\mathbf{x}^{\lambda_1^A}$ and FV $\mathbf{x}^{\lambda_2^C}$) is studied and presented in Figures 75 and 76. The graphs presented are variations on a five node graph. The first graph, (a), is an incomplete cycle, it is of interest that both eigenvectors assign the same values, to the nodes, the same occurs to the second graph (b), the complete cycle. For the third graph, (c), the differences in both eigenvectors start to occur. The PFE assigns the highest value to the node that has an in-degree of two. The other nodes get decreasing values as they become less affected by that node. The FV assigns a value of 2 to all the nodes that are within both cycles, and a value of 1 to the one node that is outside, except for the node with an out-degree of two. In the fourth graph (d), the sub-cycle is made smaller, and the PFE assigns a higher value to the node with in-degree of two than it did in graph (c). The node downstream of that receives the same centrality ranking (23%) as it did in graph (c), the node upstream of that also receives the same ranking (14%). The FV assigns the same valuation as it did for graph (c) as it does for (d), meaning that it is not affected by the size of the sub-cycles, as the PFE is. When the sub-cycle is made one node smaller, the PFE once again increases the valuation of the node with in-degree of two by 1%, and its downstream node by the same amount, the upstream node's valuation decreases by 1%. The FV once again confirms the observation that it is not affected by the length of the sub-cycles.

Graph (f) is a more complex variation of the graphs shown in (c), (d), and (e). There are now two different sub-cycles within the main cycle. The PFE assigns the same valuation to both nodes with in-degree of two, while the FV assigns a higher valuation to the node that has in-degree of two but is inside all three cycles. That node is 4 times as central in according to the FV as the node that is only part of the larger cycle. Its behavior is most likely more complex, since the dynamical interactions that can occur to the node with the value of 4 are more diverse than the node with the value of 1, which would only change as its upstream node changes. Interestingly, the PFE also assigns the lowest value to this

node, although it does not distinguish in between the other pairs of nodes.

Graphs (g) and (h) simply exemplify that as the graph becomes more complete, as long as it is perfectly symmetric, both ranking schemes will assign the same valuation to all nodes. The interesting observation from this is how $\lambda_1^{\mathcal{A}}$ has increased in value. As the cycles in graphs (c), (d), and (e) have become smaller, $\lambda_1^{\mathcal{A}}$ has consistently increased. When an additional sub-cycle was included, $\lambda_1^{\mathcal{A}}$ increased once again. The growth is super-linear, from 1.17, to 1.19, to 1.24, to 1.32. In graphs (i) and (j), the outer cycle is first removed, and then the sub-cycles are made smaller, an indication of faster dynamics. Once again, the PFE and FV assign equal valuations to all nodes since the graphs are perfectly symmetric and the interesting observation comes from studying $\lambda_1^{\mathcal{A}}$. Once the outer cycle is removed, $\lambda_1^{\mathcal{A}}$ drops to one. There is only one cycle in graph (i), it is in fact isomorphic with graph (b), and therefore it obtains the same valuation and eigenvalue. Graph (j) has sub-cycles of length two instead of three, as graph (h) did. This would indicate that the graph may display faster dynamics, but its $\lambda_1^{\mathcal{A}}$ value remains the same.

Finally, graph (k) is the union of graphs (c), (d), and (e), and graph (l) is the same graph, but instead of one node being connected to all other nodes, that one node receives input from all other nodes. The first observation of interest is that $\lambda_1^{\mathcal{A}}$ is higher for these two graphs than it is for (c), (d), (e), and (f), and more importantly, it is the same for both. The differences in PFE and FV are the most important from these two graphs. We begin with graph (k). If the PFE is normalized by its smallest value (12% in this case), the values for the nodes starting with the top node and progressing counter-clockwise, is 1.50, 1.00, 1.58, 2.00, and 2.25. Meanwhile, the FV assigned values of 1, 1, 2, 3, and 4. The FV is describing the number of cycles that go through a given node, with the exception of the top node which receives a valuation of unity. The PFE is assigning a different valuation to the nodes, where the centrality of the top node is kept more in accordance with the rest of the nodes in the sub-cycles, but where the valuation of the node in the external cycle (the one valuated at 12%) is not as reduced as it is in the FV. The most central node in the PFE is only 2.25 times more central than the least central node, while in the FV, the most central node is 4 times more central, that is a difference of 78% for relative valuation. A similar

observation can be drawn from studying graph (l). In this graph all the nodes affect the top node. The FV for the top node is now eight times larger than the value of the nodes in the outer cycle. Meanwhile, the PFE normalized by its smallest value assigns the following values to its nodes (starting with the top node and moving counter-clockwise) 5.57, 3.57, 2.43, 1.57, and 1.00. The relative valuation difference for this graph is now $\frac{8}{5.57} = 1.43$, a 43% difference, versus a 78% difference as shown in graph (k).

APPENDIX F

SOURCE CODE

F.1 Jain-Krishna Model

This MATLAB program is based on the work by Jain and Krishna.[175] It extends their work by allowing different types of networks to be created.

F.1.1 JKModel.m

```
function [t_hist,X_hist,lam1_hist,E_hist] = JKModel(s,p,r,tf,initialGraph,...
                                                    newSpeciesCatalysis,intTechnique)
% [t_hist,X_hist,lam1_hist,E_hist] = JKModel(s,p,r,tf,initialGraph,...
%                                     newSpeciesCatalysis,intTechnique)
%
% Implementation of the Jain and Krishna model to study the evolution of
% complexity in catalytic networks. [Jain and Krishna, 1998]
% Additional features have been added, including the ability to attach new
% species preferentially to the most populated species, and different
% initial topologies for the network other than random (G(p) Model).
%
% INPUTS
% s:  Species Size
% p:  Connection probability
% r:  Relative Perturbation Parameter for X
% tf: Final Time
% initialGraph: The topology of the initial network, either 'G(p)' ;
%       'Small-World' ; 'Scale-Free'
% newSpeciesCatalysis: How new species attach, either 'random' or
%       'preferential'
% intTechnique: Integration Technique, either 'integrate' ; 'steady-state'
%
% OUTPUTS
% t_hist: vector of time
```

```
25 % X_hist: time history of the state of each species
26 % lam1_hist: time history of the principal component (lambda_1)
27 % E_hist: time history of the total number of links in the network
28
29 % (c) 2008 Balestrini Robinson, Santiago
30
31 t = 0;       % Initial Time
32 T = 1;       % Integration Time Step
33
34 t_hist = [];     % Time History
35 X_hist = [];     % History of Population Size
36 E_hist = [];     % History of Total Number of Links
37 lam1_hist = []; % History of largest eigenvalue
38
39 % G(p) G(M) Small-World Scale-Free
40 %initialGraph = 'Scale-Free';
41
42 % Random or Preferential
43 %newSpeciesCatalysis = 'Random';
44
45 % Select method for integrating continuous states
46 %intTechnique = 'steady-state';
47
48 % Initialize the population
49 X = rand(s,1);
50
51 C = createCatalyticNetwork(s,p,X,initialGraph);
52
53 while t < tf
54     % Normalize the population (1-norm)
55     X = X / sum(X);
56
57     % Keep a history of the total number of edges
58     E_hist = [E_hist;sum(sum(C))];
59
```

```
60      % Calculate the eigenvectors and eigenvalues of the adjancency matrix
61      [V,D] = eig(C);
62
63      % Identify the largest value eigenvalue and its associated eigenvector
64      [lam1_hist(end+1),i] = max(real(diag(D)));
65
66      % Calculate the Dynamics either by numerically integrating or by
67      % determining the steady state value by the attractor
68      switch intTechnique
69          case 'integrate'
70              [t,X,t_hist,X_hist] = intCatalyst(t,X,C,T,10^-4,t_hist,X_hist);
71          case 'steady-state'
72              X = V(:,i);
73              X = real(X);
74              X = X / sum(X);
75              X(X<0) = 0;
76              %X(or(X<0,imag(X)) > 0) = 0;
77              X_hist = [X_hist;X'];
78              t = t+T;
79              t_hist = [t_hist;t];
80      end
81
82      % Find set of nodes with minimum value X
83      minX = find(X==min(X));
84      % Randomize their order and pick the node with the minimum random value
85      minX = sortrows([rand(length(minX),1) minX]); node = minX(1,2);
86
87      X(node) = r;
88
89      % Reconnect the selected node randomly
90      C = modifyCatalyticNetwork(C,X,node,p,newSpeciesCatalysis);
91
92      % Perturb X and ensure that all its elements remain positive
93      X = X .* ( 1 + r * ones(s,1) - r / 2);
94      X(X < 0) = 0;
```

```
95 end
96
97 figure;
98 plot(t_hist,sum(ceil(X_hist),2),'k-',t_hist,50*lam1_hist','k:');
99 title(sprintf('Initial %s Network, with %s New Species Attachment',...
100                                 initialGraph,newSpeciesCatalysis));
```

F.1.2 createCatalyticNetwork.m

```
1  function C = createCatalyticNetwork(s,p,X,randomGraphTechnique)
2
3  switch randomGraphTechnique
4      case 'G(p)'
5          % Create the catalytic network with size 's' and catalytic probability '
            p'
6          C = ceil(ones(s)*p - rand(s));
7          % Remove the diagonal elements of C to avoid self-replicating species
8          C = C - diag(diag(C));
9      case 'Scale-Free'
10         % Based on the B-A Model by Barabasi and Albert
11
12         % Check that enough links are required
13         m = ceil(s*(s-1)*p) - 2;     % Number of edges to add
14         if m < 3
15             error('Increase the value of p to at least %1.5f',3/(s*(s-1)));
16         end
17
18         pf = 2 * s * p ;
19
20         % Start with two nodes, each with in-degree 1
21         e = [1 2;2 1];
22
23         i = 3;
24         % Add the rest of the nodes
25         while m > 0 && i <= s
26             % Determine the 'popularity' of each node based on their in-degree
27             pIn = histc(e(:,2),0.5:1:i-.5);
```

```
28              pOut = histc(e(:,1),0.5:1:i-.5);
29              pIn = pIn / sum(pIn);
30              pOut = pOut / sum(pOut);
31
32              pConn = pf * ( pIn * exp(-m/10) + pOut * (1-exp(-m/10) ) );
33
34              connNodes = pConn - rand(size(pConn));
35
36              connNodes = find(connNodes > 0);
37
38              e = [e;ones(size(connNodes))*i,connNodes];
39
40              m = m - length(connNodes);
41
42              i = i + 1;
43          end
44
45          % Create adjacency matrix from the list of edges
46          C = accumarray([e(:,2),e(:,1)],1,[s,s]);
47  end
```

F.1.3 modifyCatalyticNetwork.m

```
1   function C = modifyCatalyticNetwork(C,X,k,p,attachmentTechnique)
2
3   switch attachmentTechnique
4       case 'Random'
5           % Randomly reconnect node 'k' with probability 'p'
6           C(:,k) = ceil(ones(size(C,1),1)*p - rand(size(C,1),1));
7           C(k,:) = ceil(ones(1,size(C,1))*p - rand(1,size(C,1)));
8           % Ensure that C(k,k) is 0 to avoid self replicating species
9           C(k,k) = 0;
10      case 'Preferential'
11          n = size(C,1);
12          m = p * n;
13          % Determine re-connection probabilty based on X
14          prob = X/sum(X)-rand(size(X));
```

```
15          [JUNK,I] = sort(prob,'descend');
16          C(:,k) = zeros(n,1);
17          C(k,:) = zeros(1,n);
18          C(I(1:ceil(m)),k) = ceil(.5*(m/ceil(m)) - rand(ceil(m),1));
19          C(k,I(1:ceil(m))) = ceil(.5*(m/ceil(m)) - rand(1,ceil(m)));
20          % Ensure that C(k,k) is 0 to avoid self replicating species
21          C(k,k) = 0;
22  end
```

F.1.4 intCatalyst.m

This is the integration approach to identify the state of the catalytic network. The steady-state calculation based on the PFE can be demonstrated by running the numerical integration repeatedly for a large number of different networks.

```
1  function [t,X,t_hist,X_hist] = intCatalyst(t,X,C,T,TOL,t_hist,X_hist)
2
3  OPTIONS = odeset('NonNegative',1:s);
4
5  while sum((catalysis(0,X,C)).^2) > TOL^2
6      [tdyn,Xdyn] = ode45(@catalysis,[t t+T],X,OPTIONS,C);
7
8      t_hist = [t_hist;tdyn];
9      X_hist = [X_hist;Xdyn];
10
11     % Determine the final state for X
12     X = Xdyn(size(Xdyn,1),:)';
13
14     t = t + T;
15 end
```

F.1.5 ssCatalyst.m

This is the steady state approach to identifying the state of the catalytic network based on the principal eigenvector of the adjacency matrix. Jain and Krishna use this approach to speed up the simulation process. The results of ssCatalysis.m can be verified against intCatalysis.m to ensure that the approximations are correct.

```
function [t,X,t_hist,X_hist] = ssCatalyst(t,X,C,T,t_hist,X_hist)

[V,D] = eig(C);

[JUNK,i] = max(real(diag(D)));
X = V(:,i);
X(X<0) = 0;
X_hist = [X_hist;X'];
t = t+T;
t_hist = [t_hist;t];
```

F.2 Random Boolean Network Model

F.2.1 RBN.m

This is version 9.0 of the RBN code. Its purpose is, given a set of RBN characteristics, generate networks of sufficient complexity, measure their structural characteristics, rank their nodes based on some of these structural characteristics, and compare the reductionist and holistic fidelities of each node ranking scheme and output them.

```
function [output] = RBNv9(varargin)

nettypes  = {'1-D Lattice','G(n,p)','Small-World','Scale-Free'};    % Types of
 Networks
ruletypes = {'Single','Multi'};                        % Multi-Rule or Single
 Rule
srmVec    = 1:15;
reps      = 5;
tmax      = 1000;

switch nargin
    case 0
        ID = 0;
        n = round(rand*50)+30;
        nettype = round(rand*2)+2;
        ruletype = round(rand)+1;
        p = rand*.02+0.04;
```

```
16          rd = rand*.15+0.15;
17          b = rand*0.3+0.5;
18          maxGZIPc = getMaxGZIPc(n,tmax,20);
19      case 7
20          if isstring(varargin{1})
21              ID = str2double(varargin{1});
22              n  = str2double(varargin{2});
23              nettype = str2double(varargin{3});
24              ruletype = str2double(varargin{4});
25              p = str2double(varargin{5});
26              rd = str2double(varargin{6});
27              b = str2double(varargin{7});
28          else
29              ID = round(varargin{1});
30              n  = round(varargin{2});
31              nettype = varargin{3};
32              ruletype = varargin{4};
33              p = varargin{5};
34              rd = varargin{6};
35              b = varargin{7};
36          end
37          maxGZIPc = getMaxGZIPc(n,tmax,20);
38      case 8
39          maxGZIPc = varargin{end};
40      otherwise
41          error('Incorrect number of inputs');
42  end
43
44  % Create complex RBN
45  network = createNetwork(nettypes{nettype},n,p,15);
46  RBN = createRBN(network,ruletypes{ruletype},b);
47  RBN.state = round(rand(n,1));
48  tsm_R = evolveRBN(RBN,tmax);
49
50  counter = 0;
```

```
while median(countTransitionsPerNode(tsm_R)) < 5 && counter < 50
    network = createNetwork(nettypes{nettype},n,p,15);
    RBN = createRBN(network,ruletypes{ruletype},b);
    RBN.state = round(rand(n,1));
    tsm_R = evolveRBN(RBN,tmax);
    counter = counter + 1;
end
fprintf('  %i Different RBNs created\n',counter+1);

% Compute Network Metrics
OdC = getOdC(network);
[D1,V1] = getPFE(network);
[D2,V2,nComp,D2un] = getL2(network);
dIn  = sum(network,1)';
dOut = sum(network,2);
clus = clustering_coefficients(sparse(network));
core = core_numbers(sparse(network));
nEdges = sum(sum(network));

% Structural Ranking Metric (SRM)
srm = [ V1 ...                                % (1)  PFE
        max(V1) - V1 + min(V1) ...            % (2)  Inverse PFE
        V1 + median(V1) ...                   % (3)  Corrected PFE
        V2 - min(V2) + min(V2(V2>0)) ...      % (4)  Corrected Fiedler
        abs(V2) ...                           % (5)  Absolute Fiedler
        clus ...                              % (6)  Clustering Coefficient
        max(clus) - clus + min(clus) ...      % (7)  Inverse Clustering
        core ...                              % (8)  Core Number
        max(core) - core + min(core) ...      % (9)  Inverse Core Number
        dIn ...                               % (10) In-Degree
        max(dIn)  - dIn  + min(dIn) ...       % (11) Inverse In-Degree
        dOut ...                              % (12) Out-Degree
        max(dOut) - dOut + min(dOut) ...      % (13) Inverse Out-Degree
        ones(n,1) ...                         % (14) Uniform
        rand(n,1) ];                          % (15) Random
```

```
86
87 % Remove zero-rankings from the SRM vectors
88 for i = 1:size(srm,2)
89     if isempty(srm((srm(:,i)>0),i))
90         srm(:,i) = ones(size(srm(:,i)));
91     end
92     srm(srm(:,i) == 0,i) = srm(srm(:,i) == 0,i) + min(srm((srm(:,i)>0),i)) / 10;
93 end
94
95 srm = srm ./ repmat(sum(srm,1),n,1);         % 1-norm of SRM
96
97 % figure;plot(sort(srm),'LineWidth',2);
98 % legend('p(PFE)','i(PFE)','c(PFE)','c(FV)','a(FV)','p(CC)','i(CC)','p(CN)','i(
    CN)','p(ID)','i(ID)','p(OD)','i(OD)','U','R');
99
100 nodes0 = sum(srm == 0,1);
101 meanRank = mean(srm,1);
102 medianRank = median(srm,1);
103
104 rulesActive = sum(2 .^ RBN.inDeg);
105
106 for j = 1:reps
107     decimal_R = getDecimal(tsm_R)';
108     hammdist_R = getHammdist(tsm_R);
109     power_R = sum(tsm_R,1)/n;
110
111     [DECfR,DECaR] = getSpectrum(decimal_R);
112     [HMDfR,HMDaR] = getSpectrum(hammdist_R);
113     [POWfR,POWaR] = getSpectrum(power_R);
114
115     % Compute Complexity Metrics
116     GZIPcomplexity_R(1,j) = getGZIPc(tsm_R,maxGZIPc);
117     LZCcomplexity_R(1,j) = getLZC(decimal_R);
118     TSMentropy_R(1,j) = entropy(tsm_R');
119
```

```
120     nodeActy_R = countTransitionsPerNode(tsm_R);
121     SAN_R = find(nodeActy_R > median(nodeActy_R));
122     nSAN_R(1,j) = length(SAN_R);
123
124     NC_R = getNodeCorrelation(tsm_R);
125     nNC_R(1,j) = size(NC_R,1);
126
127     for k = 1:length(srmVec)
128         [tsm_M,rulesNeglected(k,j)] = evolveRBN(RBN,tmax,srm(:,srmVec(k)),rd);
129
130         decimal_M = getDecimal(tsm_M)';
131         hammdist_M = getHammdist(tsm_M);
132         power_M = sum(tsm_M,1)/n;
133
134         % Compute Complexity Metrics
135         GZIPcomplexity_M(k,j) = getGZIPc(tsm_M,maxGZIPc);
136         LZCcomplexity_M(k,j) = getLZC(decimal_M);
137         TSMentropy_M(k,j) = entropy(tsm_M');
138
139         nodeActy_M = countTransitionsPerNode(tsm_M);
140         SAN_M = find(nodeActy_M > median(nodeActy_M));
141         nSAN_M(k,j) = length(SAN_M);
142
143         NC_M = getNodeCorrelation(tsm_M);
144         nNC_M(k,j) = size(NC_M,1);
145
146         % Compute Binary Fidelity
147         BINfid(k,j) = getBINfidelity(tsm_R,tsm_M,30);
148
149         % Compute Spectrum Metrics
150         [DEC_FP(k,j),DEC_FN(k,j),DECfid(k,j)] = getSpectrumErr(DECfR,DECaR,
              decimal_M);
151         [HMD_FP(k,j),HMD_FN(k,j),HMDfid(k,j)] = getSpectrumErr(HMDfR,HMDaR,
              hammdist_M);
```

```
152         [POW_FP(k,j),POW_FN(k,j),POWfid(k,j)] = getSpectrumErr(POWfR,POWaR,
             power_M);
153
154         SAN_FP(k,j) = length(setdiff(SAN_M,SAN_R)) / (n-nSAN_R(j));
155         SAN_FN(k,j) = length(setdiff(SAN_R,SAN_M)) / (nSAN_R(j));
156
157         NC_FP(k,j) = size(setdiff(NC_M,NC_R,'rows'),1) / ( 0.5*n*(n-1) - nNC_R(j
             ));
158         NC_FN(k,j) = size(setdiff(NC_R,NC_M,'rows'),1) / nNC_R(j);
159     end
160     % Choose a different Initial Condition and Evolve the RBN
161     RBN.state = round(rand(n,1));
162     tsm_R = evolveRBN(RBN,tmax);
163 end
164
165 stdout = [
166     ID;
167     n;
168     nettype;
169     ruletype;
170     p;
171     rd;
172
173     OdC;
174     D1;
175     D2;
176     D2un;
177     nComp;
178     nEdges;
179     mean(GZIPcomplexity_R);
180     mean(LZCcomplexity_R);
181     mean(TSMentropy_R);
182     rulesActive;
183     mean(nSAN_R) / n;
184     mean(nNC_R) / (0.5*n*(n-1))];
```

```
185
186 for i = 1:length(srmVec)
187     meanSAN_FP(i,1) = mean(SAN_FP(i,¬isnan(SAN_FP(i,:))));
188     meanSAN_FN(i,1) = mean(SAN_FN(i,¬isnan(SAN_FN(i,:))));
189     meanNC_FP(i,1) = mean(NC_FP(i,¬isnan(NC_FP(i,:))));
190     meanNC_FN(i,1) = mean(NC_FN(i,¬isnan(NC_FN(i,:))));
191 end
192
193 output = [
194     repmat(ID,1,length(srmVec));
195
196     srmVec;
197
198     repmat(stdout(2:end),1,length(srmVec));
199
200     nodes0/n;
201     meanRank;
202     medianRank;
203     mean(rulesNeglected,2)';
204
205     mean(GZIPcomplexity_M,2)';
206     mean(LZCcomplexity_M,2)';
207     mean(TSMentropy_M,2)';
208     mean(nSAN_M,2)' / n;
209     mean(nNC_M,2)'  / (0.5*n*(n-1));
210
211     % Reductionist Fidelity
212     mean(BINfid,2)';
213     mean(DECfid,2)';
214     mean(HMDfid,2)';
215     mean(POWfid,2)';
216
217     % Holistic Fidelity
218     1 - 0.5 * ( mean(DEC_FP,2) / max([mean(DEC_FP,2);1]) + mean(DEC_FN,2) / max
        ([mean(DEC_FN,2);1]) )';
```

```
219     1 - 0.5 * ( mean(HMD_FP,2) / max([mean(HMD_FP,2);1]) + mean(HMD_FN,2) / max
          ([mean(HMD_FN,2);1]) )';
220     1 - 0.5 * ( mean(POW_FP,2) / max([mean(POW_FP,2);1]) + mean(POW_FN,2) / max
          ([mean(POW_FN,2);1]) )';
221     1 - 0.5 * ( meanSAN_FP     / max([meanSAN_FP;1])     + meanSAN_FN     / max
          ([meanSAN_FN;1])     )';
222     1 - 0.5 * ( meanNC_FP      / max([meanNC_FP;1])      + meanNC_FN      / max
          ([meanNC_FN;1])      )';
223     ]';
```

F.2.2 createNetwork.m

Creates either a 1-D Lattice, an Erdös-Rényi G(n,p), a Watts-Strogatz Small-World, or Albert-Barabási Scale-Free network.

```
1  function [network] = createNetwork(type,n,p,varargin)
2
3  if nargin == 5
4      maxInDeg = varargin{1};
5      selfloop = varargin{2};
6  else
7      maxInDeg = 10;
8      selfloop = 0;
9  end
10
11 switch type
12     case '1-D Lattice'
13         C = diag(ones(n,1));
14         C = C + [zeros(n-1,1), diag(ones(n-1,1)); zeros(1,n)];
15         C = C + triu(C,1)';
16         C(1,n) = 1;
17         C(n,1) = 1;
18         network = C;
19     case 'G(n,p)'
20         % Create the network with size 'n' and probability 'p'
21         C = ceil(p - rand(n));
```

```
22          % Remove the C(i,i) elements to avoid loops
23          network = C - diag(diag(C));
24          inDeg = sum(network,1)';
25          ni = find(inDeg > maxInDeg);
26          for i = 1:length(ni)
27              nrem = inDeg(ni(i)) - maxInDeg;
28              links = find(network(:,ni(i)) > 0);
29              links = sortrows([rand(size(links)) links],1);
30              links = links(1:nrem,2);
31              network(links,ni(i)) = 0;
32          end
33          if selfloop
34              network = network + diag(ones(n,1));
35          end
36      case 'Small-World'
37          % Create lattice network
38          m = n*(n-1)*p;
39          k = ceil(0.5*m/n);
40          k = min([k,4]);
41          C = zeros(n);
42          for i = 1:k
43              C = C + [zeros(n-i,i), diag(ones(n-i,1)) + [zeros(i,n-2*i),diag(ones
                (i,1)); zeros(n-2*i,n-i)] ; zeros(i,n)];
44          end
45          C = C + triu(C,1)';
46          edges = find(C);
47          edges = sortrows([rand(size(edges)) edges],1);
48          edges = edges(:,2);
49          for i = 1:round(0.03*length(edges))
50              C(edges(i)) = 0;
51              C(ceil(rand*n),ceil(rand*n)) = 1;
52          end
53          edges = find(C);
54          edges = sortrows([rand(size(edges)) edges],1);
55          edges = edges(:,2);
```

```
56          ee = round(length(edges) - m);
57          if ee < 0
58              noedges = find(C==0);
59              noedges = sortrows([rand(size(noedges)) noedges],1);
60              noedges = noedges(:,2);
61              C(noedges(1:-ee)) = 1;
62          elseif ee > 0
63              C(edges(1:ee)) = 0;
64          end
65          C = C - diag(diag(C));
66          network = C;
67          if selfloop
68              network = network + diag(ones(n,1));
69          end
70      case 'Scale-Free'
71          % Based on the B-A Model by Barabasi and Albert
72
73          % Check that enough links are required
74          m = ceil(n*(n-1)*p) - 2;    % Number of edges to add
75          if m < 6
76              error('Increase the value of p to at least %1.5f',6/(n*(n-1)));
77          end
78
79          seed = [0 1 0 0 1;1 0 0 1 0;0 0 0 1 0;0 1 1 0 0;1 0 0 0 0];
80          i = size(seed,1);
81          mlinks = p*n*(n-1);
82
83          network = zeros(n,n);
84          network(1:i,1:i) = seed;
85          links = sum(sum(network));
86          connectedNetwork = 0;
87          while links < mlinks
88              while i < n
89                  i = i+1;
90                  deg = sum(network,1);
```

```
91                  deg(i) = -1;
92                  deg(deg >= maxInDeg) = -1;
93                  p = (deg+2)/links;
94                  [nodesToLinkTo] = find(p - rand(1,n) > 0);
95                  vec = sortrows([rand(length(nodesToLinkTo),1) nodesToLinkTo']);
96                  if size(vec,1) > 0 && network(i,vec(1,2)) == 0
97                      network(i,vec(1,2)) = 1;
98                      links = links + 1;
99                  end
100                 if links / mlinks > 0.95 && connectedNetwork == 0
101                     % Ensure that every node has at least one incoming
                          connection
102                     disconnectedNodes = find(sum(network,2)==0);
103                     for i = 1:length(disconnectedNodes)
104                         network(ceil(rand*n),disconnectedNodes(i)) = 1;
105                     end
106                     connectedNetwork = 1;
107                 end
108                 if links >= mlinks
109                     break
110                 end
111             end
112             i = 1;
113         end
114         if selfloop
115             network = network + diag(ones(n,1));
116         end
117     otherwise
118         error('TYPE must either be ''1-D Lattice'', ''G(n,p)'', ''Small-World'',
              or ''Scale-Free''')
119 end
120 network(network > 1) = 1;
```

F.2.3 getOdC.m

This function is based on the off-diagonal complexity metric proposed by J.C. Claussen to efficiently estimate the complexity of a graph. Code is based on the algorithm presented in [85].

```
function [OdC] = getOdC(g)
% [OdC] = getOdC(g) where g is the adjacency matrix of a graph g
% Based on the algorithm by J.C. Claussen [Physica A 375 (2007) 365 373]

g = g ≠ 0;              % Ensure that the adjacency matrix is binary
l = sum(g,1);            % Degree Distribution
c = zeros(max(l));       % Preallocated for speed
a = zeros(size(g,1));    % Preallocated for speed

for m = 1:max(l)
    for n = m:max(l)
        i = find(l==m);
        j = find(l==n);
        if ¬isempty(i) && ¬isempty(j)
            c(m,n) = sum(sum(g(i,j)));
        end
    end
end

for k = 1:length(l)
    a(k) = sum(diag(c,k-1));
end

A = sum(a);

a = a/A;

h = a.*log(a);
h(isnan(h)) = 0; % 0*ln(0) is understood to be equal to 0 and not NaN

```

```
31 OdC = -sum(h);
```

F.2.4 createRBN.m

```
1  function [RBN] = createRBN(varargin)
2
3  switch nargin
4      case 1
5          A = varargin{1};
6          rules = 'Single';
7          ruleBias = 0.5;
8      case 2
9          A = varargin{1};
10         rules = varargin{2};
11         ruleBias = 0.5;
12     case 3
13         A = varargin{1};
14         rules = varargin{2};
15         ruleBias = varargin{3};
16     otherwise
17         error('Incorrect number of inputs');
18 end
19
20 n = size(A,1);
21
22 RBN.state = false(n,1);
23 RBN.inDeg = sum(A,1)';
24
25 m = max(RBN.inDeg);
26
27 RBN.inputs = zeros(n,m);
28 for i = 1:n
29         RBN.inputs(i,m-RBN.inDeg(i)+1:m) = find(A(:,i))';
30 end
31
32 switch rules
```

```
33      case 'Single'
34          rule = rand(1,2^m) > ruleBias;
35          RBN.rules = repmat(rule,n,1);
36      case 'Multi'
37          RBN.rules = rand(n,2^m) > ruleBias;
38  end
39
40  for i = 1:n
41     RBN.rules(i,RBN.inDeg(i)^2+1:end) = 0;
42  end
43
44  RBN.inputs = sparse(RBN.inputs);
```

F.2.5 evolveCRBNfast.m

This function is an adaptation of `evolveCRBN` included in the RBN toolbox by Christian Schwarzer. It evolves an RBN using the classical RBN scheme two orders of magnitude faster than the original function.

```
1  function [varargout] = evolveCRBNfast(node,varargin)
2  % EVOLVECRBNFAST Based on evolveCRBN, but executes 2 orders of magnitude faster.
3  % Develops the network gradually K discrete time-steps according to CRBN
4  % (Classical Random Boolean Network) update scheme.
5  %
6  %   EVOLVECRBNFAST(NODE) advances all nodes in NODE one time-step in CRBN update
     mode.
7  %
8  %   EVOLVECRBNFAST(NODE, K) advances all nodes in NODE K time-steps in CRBN
     update mode.
9  %
10 %   EVOLVECRBNFAST(NODE, K, TK) advances all nodes in NODE K time-steps in CRBN
     update mode
11 %   and saves all TK steps all node-states and the timeStateMatrix to the disk.
12 %
13 %
14 %   Input:
15 %       node              - 1 x n structure-array containing node information
```

```
16 %       k                  - (Optional) Number of time-steps
17 %       tk                 - (Optional) Period for saving node-states/
    timeStateMatrix to disk.
18 %
19 %
20 %   Output:
21 %       nodeUpdated        - 1 x n sturcture-array with updated node information
22 %                            ("lineNumber", "state", "nextState")
23 %       timeStateMatrix    - n x k+1 matrix containing calculated
24 %       time-state evolution
25
26 %   Author: Santiago Balestrini - ASDL Georgia Tech [based on the RBN Toolbox by
     Christian Schwarzer]
27 %   CreationDate: 13.10.2008 LastModified: 01.15.2009
28
29 switch nargin
30     case 1
31         k = 1;
32         tk = inf;
33     case 2
34         k = varargin{1};
35         tk = inf;
36     case 3
37         k = varargin{1};
38         tk = varargin{2};
39     otherwise
40         error('Incorrect number of inputs. Refer to HELP evolveCRBNfast for more
              details.');
41 end
42
43 n = length(node);
44
45 initState = zeros(n+1,1);
46 for i = 1:n
47     initState(i) = node(i).state;
```

```
48 end
49
50 rules = getRulesMAT(node);
51 inputs = getInputMAT(node);
52
53 tsm = zeros(n+1,k+1);
54 tsm(:,1) = initState;
55
56 m = size(inputs,2);
57
58 rules = [rules;repmat(-1,1,size(rules,2))];
59 pow2mat = repmat(pow2(m-1:-1:0),n+1,1);
60
61 for t = 2:k+1
62     states = reshape(tsm(inputs,t-1),n+1,m);
63     nextStateIndex = sum(states .* pow2mat,2) + 1;
64     tsm(1:n+1,t) = rules(sub2ind(size(rules),1:n+1,nextStateIndex'));
65     tsm(tsm(:,t)==-1,t) = tsm(tsm(:,t)==-1,t-1);
66 end
67
68 varargout{1} = logical(tsm(1:n,:));
69
70 function [rulesMAT] = getRulesMAT(node)
71
72 n = length(node);
73
74 ruleLength = zeros(n,1);
75
76 for i = 1:n
77     ruleLength(i) = length(node(i).rule);
78 end
79
80 m = max(ruleLength);
81 rulesMAT = repmat(-1,n,m);
82
```

```
for i = 1:n
    rulesMAT(i,1:ruleLength(i)) = node(i).rule';
end
function [inputMAT] = getInputMAT(node)
n = length(node);
indeg = zeros(1,n);

for i = 1:n
    indeg(i) = length(node(i).input);
end

m = max(indeg);

inputMAT = repmat(n+1,n+1,m);

for i = 1:n
    inputMAT(i,m-indeg(i)+1:m) = node(i).input;
end
```

F.2.6 evolveRBN.m

```
function [varargout] = evolveRBN(RBN,varargin)
% Develops the network gradually K discrete time-steps according to CRBN
% (Classical Random Boolean Network) update scheme.
%
%   EVOLVECRBNFAST(RBN) advances all nodes in NODE one time-step in CRBN update mode.
%
%   EVOLVECRBNFAST(RBN, K) advances all nodes in NODE K time-steps in CRBN update mode.
%
%   EVOLVECRBNFAST(RBN, K, F) advances all nodes in NODE K time-steps in CRBN update mode
%   with F fidelity for each node.
%
```

```
12 %   EVOLVECRBNFAST(RBN, K, F, RD) advances all nodes in NODE K time-steps in
   CRBN update mode
13 %   with F fidelity for each node, with a fractional random error of RD for not
   changing state.
14 %
15 %   EVOLVECRBNFAST(RBN, K, F, RD, FLAG) advances all nodes in NODE K time-steps
   in CRBN update mode
16 %   with F fidelity for each node, with a fractional random error of RD for
17 %   not changing state if FLAG is FALSE, or switching the state to the
18 %   opposite if FLAG is TRUE
19 %
20 %   Input:
21 %       node              - 1 x n structure-array containing node information
22 %       k                 - (Optional) Number of time-steps
23 %       f                 - (Optional) Fidelity per node n x 1 vector ranging
   between 0 and 1
24 %       rd                - (Optional) Fraction of rules to discard
25 %       flag              - (Optional) (F) Error keeps state as previous
26 %                                      (T) Error makes state the opposite
27 %
28 %
29 %   Output:
30 %       timeStateMatrix   - n x k+1 matrix containing calculated time-state
   evolution
31
32 %   Author: Santiago Balestrini - ASDL Georgia Tech
33 %   CreationDate: 04.02.2009 LastModified: 04.02.2009
34
35 n = length(RBN.state);
36
37 switch nargin
38     case 1
39         k = 1;
40         f = ones(n,1);
41         rd = 0;
```

```
42          opposite = 0;
43          mal = 0;
44      case 2
45          k = varargin{1};
46          f = ones(n,1);
47          rd = 0;
48          opposite = 0;
49          mal = 0;
50      case 3
51          k = varargin{1};
52          f = varargin{2};
53          rd = 0;
54          opposite = 0;
55          mal = 0;
56      case 4
57          k = varargin{1};
58          f = varargin{2};
59          rd = varargin{3};
60          opposite = 0;
61          mal = 0;
62      case 5
63          k = varargin{1};
64          f = varargin{2};
65          rd = varargin{3};
66          opposite = varargin{4};
67          mal = 0;
68      case 6
69          k = varargin{1};
70          f = varargin{2};
71          rd = varargin{3};
72          opposite = varargin{4};
73          mal = varargin{5};
74      otherwise
75          error('Incorrect number of inputs. Refer to HELP evolveRBN for more
              details.');
```

```
76 end
77
78 m = max(RBN.inDeg);
79
80 rules = RBN.rules;
81 inputs = full(RBN.inputs);
82 inMat = inputs > 0;
83 inputs(inputs == 0) = n;
84 pow2mat = repmat(pow2(m-1:-1:0),n,1);
85 if rd > 0
86     [ignore,i] = sort(rand(n,k) .* repmat(f,1,k),1,'ascend');
87     fmat = zeros(n,k);
88     i = i(1:round(n*rd),:);
89     for j = 1:k
90         fmat(i(:,j),j) = 1;
91     end
92     fmat = fmat > 0;
93 else
94     fmat = false(n,k);
95 end
96
97 tsm = zeros(n,k+1);
98 tsm(1:n,1) = RBN.state;
99
100 if opposite
101     for t = 2:k+1
102         states = reshape(tsm(inputs,t-1),n,m);
103         nextStateIndex = sum(inMat .* states .* pow2mat,2) + 1;
104         tsm(:,t) = rules(sub2ind(size(rules),1:n,nextStateIndex'));
105         tsm(fmat(:,t-1),t) = abs(tsm(fmat(:,t-1),t)-1);
106     end
107 else
108     for t = 2:k+1
109         states = reshape(tsm(inputs,t-1),n,m);
110         nextStateIndex = sum(inMat .* states .* pow2mat,2) + 1;
```

```
111         tsm(:,t) = rules(sub2ind(size(rules),1:n,nextStateIndex'));
112         tsm(fmat(:,t-1),t) = tsm(fmat(:,t-1),t-1);
113     end
114 end
115
116 varargout{1} = logical(tsm(1:n,:));
117
118 switch nargout
119     case 2
120         varargout{2} = sum(sum(fmat(1:n,:)))/(n*k);
121     case 3
122         varargout{2} = sum(sum(fmat(1:n,:)))/(n*k);
123         varargout{3} = sum(fmat(1:n,:),2)/k;
124 end
```

F.2.7 countTransitionsPerNode.m

```
1  function nbChangesPerNode = countTransitionsPerNode(tsm)
2
3  % COUNTTRANSITIONSPERNODE Count number of changes of a node through evolution.
4  %
5  %   COUNTTRANSITIONSSPERNODE(TSM) counts for all nodes in TSM how many times
     they have changed state
6  %   over all discrete timesteps.
7  %
8  %   Input:
9  %       tsm                 - n x k+1 matrix containing node-states for n nodes
     at k timesteps
10 %
11 %   Output:
12 %       nbChangesPerNode    - Number of changes of each node (column vector)
13 %
14
15 %   Author: Santiago Balestrini, comments based on Christian Schwarzer's code
16
17 if(nargin == 1)
18     nbChangesPerNode = sum((tsm(:,1:end-1)-tsm(:,2:end))~=0,2);
```

```
19 else
20     error('Wrong number of arguments. Type: help countTransitionsPerNode');
21 end
```

F.2.8 getDecimal.m

```
1 function [tsmND] = getDECtsm(tsm)
2
3 [ignore,ni] = sort(sum(tsm,2),'descend');
4 tsmND = gray2nd(tsm(ni,:),size(tsm,1));
```

F.2.9 gray2nd.m

```
1 function [tsmND] = gray2nd(tsm,n)
2 tsmND = bin2nd(gray2bin(tsm',n),n);
```

F.2.10 gray2bin.m

```
1 function popBin = gray2bin(popGray,bits)
2
3 for i = 1:length(bits)
4     pos = sum(bits(1:i))-bits(1);
5     popBin(:,pos+1) = popGray(:,pos+1);
6     for j=2:bits(i)
7         popBin(:,pos+j) = xor(popGray(:,pos+j),popGray(:,pos+j-1));
8     end
9 end
```

F.2.11 bin2nd.m

```
1 function popND = bin2nd(popBin,bits)
2 s = size(popBin);
3 popND = sum(double(popBin).*(repmat(pow2(bits-1:-1:0),s(1),1)),2)/2^bits;
```

F.2.12 getHammdist.m

```
1 function [hd] = hammdist(mat)
2 % HAMMDIST Hamming Distance of a Binary Matrix
3 %
4 % HAMMDIST(MAT) returns the normalized Hamming distance of the binary
5 % matrix MAT for every interval of its 2nd dimension.
6 %
```

```
% The Hamming distance is normalized by its 1st dimension.

hd = sum(mat ≠ repmat(mat(:,1),1,size(mat,2)),1)/size(mat,1);
```

F.2.13 getLZC.m

This function was an adaptation of `kolmogorov(s)` by Stephen Faul based on the work of Kaspar and Schuster [186]. The definition of the normalizing parameter b has value $n/\log_2(n)$ as defined by Lempel and Ziv [209]. In reality, the maximum value of $c(s)$ is n, and therefore, that is the reason that $c(s)$ is normalized by n.

```
function [LZC] = getLZC(s)
% DATE: 9th Feb 2005
% AUTHOR: Stephen Faul (stephenf@rennes.ucc.ie)
%
% Function for estimating the Kolmogorov Complexity as per:
% "Easily Calculable Measure for the Complexity of Spatiotemporal Patterns"
% by F Kaspar and HG Schuster, Physical Review A, vol 36, num 2 pg 842
%
% Input is a digital string, so conversion from signal to a digital stream
% must be carried out a priori
%
% Lempel–Ziv Complexity (LZC) is reported a la Lempel and Ziv
% (IEEE Trans Inf Theory 75 (1976)) divided by log2(n)
% where c(n) is the kolmogorov complexity and h(n) is a normalised measure
% of complexity.

n = length(s);
c = 1;
l = 1;

i = 0;
k = 1;
k_max = 1;
stop = 0;

while stop == 0
```

```
27      if s(i+k) ≠ s(l+k)
28          if k > k_max
29              k_max = k;
30          end
31          i = i + 1;
32          if i == l
33              c = c + 1;
34              l = l + k_max;
35              if l+1 > n
36                  stop = 1;
37              else
38                  i = 0;
39                  k = 1;
40                  k_max = 1;
41              end
42          else
43              k = 1;
44          end
45      else
46          k = k + 1;
47          if l+k > n
48              c = c + 1;
49              stop = 1;
50          end
51      end
52  end
53
54  LZC = c/n;
```

F.2.14 getGZIPc.m

This function computes the normalized minimal information required to reproduce a TSM using the GNU ZIP compression algorithm. The amount of information required is normalized between the maximum (the amount required to reproduce a random TSM) and the

minimum (the amount of information required to reproduce the simplest TSM). The number of random TSMs generated to compute the maximum is set to 50, but a normalizing value can be provided to reduce the computational time of the algorithm.

```
function [GZIPc] = getGZIPc(tsm,varargin)

tsm = logical(tsm);

GZIPmin = length(dzip(logical(zeros(size(tsm)))));

switch nargin
    case 1
        GZIPmax = zeros(1,50);
        for i = 1:50
            GZIPmax(i) = length(dzip(logical(round(rand(size(tsm))))));
        end
        GZIPmax = max(GZIPmax);
    case 2
        GZIPmax = varargin{1};
    otherwise
        error('Only one or two inputs are allowed in ''getTSMgzipComplexity''');
end

[ignore,sn] = sort(sum(tsm,2));

tsm = tsm(sn,:);

GZIPc = min([size(dzip(tsm),1),size(dzip(tsm'),1)]);

GZIPc = (GZIPc - GZIPmin) / (GZIPmax - GZIPmin);
```

F.2.15 getBINfidelity.m

```
function [fidelity] = getBINfidelity(tsmR,tsmM,r)

```

```
fidelity = sum(sum(repmat(exp(-(0:size(tsmR,2)-2)/r),size(tsmR,1),1).*(tsmR(:,2:
  end)==tsmM(:,2:end))));

fidelity = fidelity/sum(sum(repmat(exp(-(0:size(tsmR,2)-2)/r),size(tsmR,1),1)));
```

F.2.16 getNodeCorrelation.m

The default setting in the MATLAB function `corr.m` is Pearson's linear correlation. A *pval* lower than 0.05 is deemed to be indication that the correlation is sufficiently greater than zero.

```
function [nodeCorr] = getNodeCorrelation(tsm)

[ignore,pval] = corr(double(tsm'));
[nodeCorr(:,1),nodeCorr(:,2)] = find(triu(pval<0.05,1));
```

F.2.17 getSpectrumErr.m

```
function [FP,FN,aR2] = getSpectrumErr(fR,aR,data)

df = fR(2);

[fM,aM] = getSpectrum(data);

if sum(fM-fR) ~= 0
    fprintf('  Unmatched Frequency Domains, error: %1.3f\n',sum(fM-fR));
end

% d(a)/d(f) for the Reference and Model RBNs
dadfR = (aR(3:end)-aR(2:end-1))/df;
dadfM = (aM(3:end)-aM(2:end-1))/df;

% Identify the indeces of the peaks
pR = find(dadfR(1:end-1) > 0 & dadfR(2:end) < 0) + 2;
pM = find(dadfM(1:end-1) > 0 & dadfM(2:end) < 0) + 2;

% False Positive and False Negative Peak Identification
% peaksR = fR( pR);
```

```
% peaksM = fM( pM);
% pFP = length(setdiff(peaksM,peaksR));
% pFN = length(setdiff(peaksR,peaksM));

% Weighted False Positive and False Negative Peak Error
FP = abs(aR(pM)-aM(pM))./aM(pM) / length(pM);
FN = abs(aR(pR)-aM(pR))./aR(pR) / length(pR);
FP = sum(FP(~isnan(FP) & abs(FP) < Inf));
FN = sum(FN(~isnan(FN) & abs(FN) < Inf));

% Amplitude R^2
RSS = (log(aR)-log(aM)).^2;
RSS = RSS(abs(RSS) < Inf);
TSS = sum((log(aR(aR>0)- mean(log(aR(aR>0))))).^2);
aR2 = 1 - sum(RSS(RSS<Inf))/sum(TSS(TSS<Inf));

% aR2df = 1 - sum((log(dadfR)-log(dadfM)).^2)/sum((log(dadfR)-mean(log(dadfR))).
  ^2);
% if rand > 0.75
%     figure;
%     subplot(3,1,1);loglog(aR(1:end),aM(1:end),'.');axis square;title(sprintf('
  R^2 = %2.1f%%',aR2*100));
%     subplot(3,1,2);loglog(dadfR,dadfM,'.');axis square;title(sprintf('R^2 = %2
  .1f%%',aR2df*100));
%     subplot(3,1,3);semilogy(fR(1:end),aR(1:end),'k',fM(1:end),aM(1:end),'r');
  hold on;
% end
```

F.2.18 getSpectrum.m

```
function [F,A] = getSpectrum(data)

tmax = length(data);

NFFT = 2^nextpow2(tmax);
y = fft(data,NFFT)/tmax;

```

```
F = 1/2*linspace(0,1,NFFT/2+1);
A = 2*abs(y(1:NFFT/2+1));
```

F.2.19 getRegret.m

```
function [regret] = getRegret(data,bz,vars,maxv)

% Ensure that the data ranges between zero and one
data(:,vars) = data(:,vars) -  repmat(min( [ data(:,vars); zeros(1,length(vars))
   ] ),size(data,1),1);
data(:,vars) = data(:,vars) ./ repmat(max( [ data(:,vars);  ones(1,length(vars))
   ] ),size(data,1),1);

regret = zeros(size(data,1),length(vars));

for i = 1:bz:size(data,1)
    for j = 1:length(vars)
        if maxv(j)
            maxval = max(data(i:i+bz-1,vars(j)));
            regret(i:i+bz-1,j) = (maxval - data(i:i+bz-1,vars(j)))/maxval;
        else
            minval = min(data(i:i+bz-1,vars(j)));
            regret(i:i+bz-1,j) = (data(i:i+bz-1,vars(j)) - minval)/(1-minval);
        end
    end
end
```

F.3 1-D Cellular Automata Complexity Study Framework

This program was based on the code contributed by Daniel de Souza Carvalho at wolfram.com.[1] It was extended to include the spectrum of the CA sequence. Figure 77 shows how the different characteristics of the CA can be studied in an interactive manner.

```
Manipulate[
 SeedRandom[rseed]; Pane[Grid[{
    {ArrayPlot[CellularAutomaton[rule, RandomInteger[1, size], size],
      Frame -> None, ColorRules -> {0 -> White, 1 -> Black},
```

[1] http://demonstrations.wolfram.com/PowerSpectrumOfCellularAutomatonDynamics/

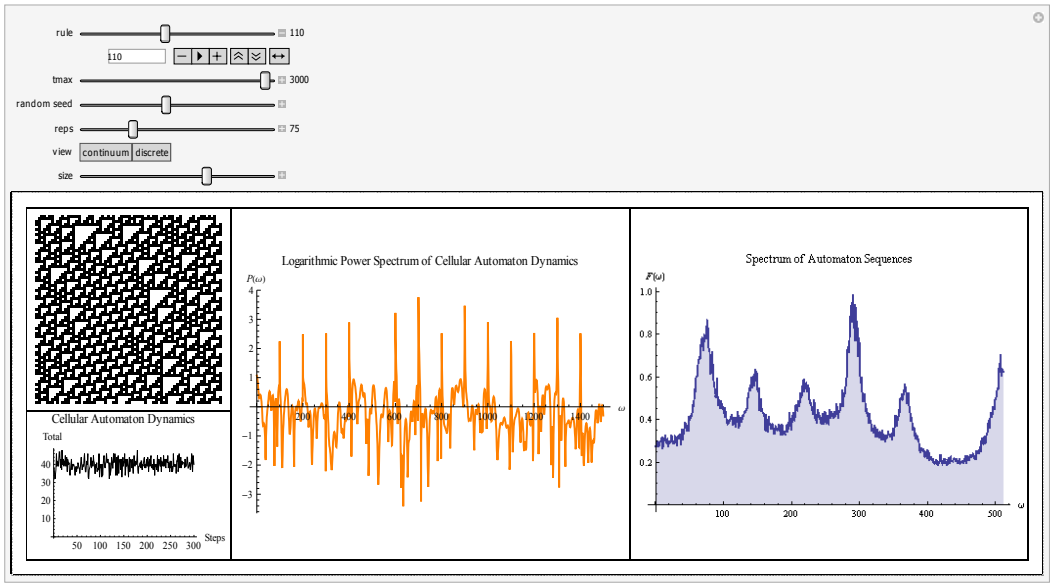

Figure 77: 1-D Cellular Automata Complexity Study Framework.

```
5       ImageSize -> { 200, Automatic}],
6      view[
7       Drop[Take[
8         Log[Abs[
9           Fourier[
10           caDynamic =
11            Total[Transpose[
12              ca = CellularAutomaton[rule, RandomInteger[1, size],
13                tmax]]]]]], {1, Floor[tmax/2]}], 1],
14      PlotRange -> All, PlotStyle -> {Thick, Orange},
15      AxesLabel -> {\[Omega], P[\[Omega]]},
16      PlotLabel ->
17       "Logarithmic Power Spectrum of Cellular Automaton Dynamics",
18      AxesOrigin -> {1, 0}, ImageSize -> {400, Automatic} ,
19      Filling -> If[view === ListPlot, Axis, None],
20      ImagePadding -> {{15, 15}, {0, 35}}],
21     view[
22      Apply[Plus,
23        Transpose[
24         Table[Take[
25           Abs[Fourier[
26             Last[CellularAutomaton[rule,
27               Table[Random[Integer], {1024}], 200 ]]]], {2,
28            1024/2}], {reps}]], {1}]/reps, PlotRange -> All,
29      PlotStyle -> {Thick}, AxesLabel -> {\[Omega], F[\[Omega]]},
30      PlotLabel -> "Spectrum of Automaton Sequences",
31      AxesOrigin -> {1, 0}, ImageSize -> {400, Automatic} ,
32      Filling -> Axis, ImagePadding -> {{15, 15}, {5, 35}}]
33     },
34    {view[Take[caDynamic, {1, Min[300, tmax]}],
35      PlotRange -> Automatic, PlotStyle -> {Thin, Black},
36      AxesLabel -> {"Steps", "Total"},
37      PlotLabel -> "Cellular Automaton Dynamics",
38      AxesOrigin -> {0, 0},
39      Filling -> If[view === ListPlot, Axis, None],
```

```
      ImageSize -> {200, Automatic},
      ImagePadding -> {{20, 30}, {20, 35}}], SpanFromAbove}},
   Frame -> All, Alignment -> {Top, Center}], {Automatic,
   Automatic}],
 {{rule, 110}, 0, 255, 1, Appearance -> "Labeled"},
 {{tmax, 3000}, 100, 3000, 1, Appearance -> "Labeled"},
 {{rseed, 123, "random seed"}, 1, 123123123, 1},
 {{reps, 40}, 1, 300, 1, Appearance -> "Labeled"},
 {{view, ListLinePlot}, {ListLinePlot -> "continuum",
   ListPlot -> "discrete"}},
 {{size, 70}, 10, 100, 1}, TrackedSymbols -> Manipulate]
```

F.3.1 ca.m

This MATLAB function uses the adapted RBN toolbox to create a 1-D cellular automata.

```
function [tsm] = ca(rule,n,initStates,tmax)

rule = nd2bin(rule/255,8);
rule = rule(end:-1:1);

net = createNetwork('1-D Lattice',n,1);
conn = initConnections(net);

node = initNodes(n,initStates,zeros(n,1),zeros(n,1));
node = assocNeighbours(node,conn);

for j = 1:n
    node(j).rule = rule;
end

tsm = evolveCRBNfast(node,tmax);
```

F.4 DiMA

These functions were the implementation of DiMA as described in Algorithm 4.1.

F.4.1 dima.m

```
function [varargout] = DiMA(varargin)
%DIMA   Executes the Digraph Modeling for Architectures (DiMA) process.
%   N = DIMA(EGM,CM,FSM,R) returns a structure with the network metrics for
%   a given Engagement Generation Matrix (EGM), a Capability Matrix (CM),
%   Force Structure Matrix (FSM), and number of stochastic repetitions R.

fprintf(' Running DiMA analysis\n');
EGM = varargin{1};
CM = varargin{2};
FSM = varargin{3};
nk = varargin{4};
out = [];
s = FSM(:,1);
f = FSM(:,2);
n = length(f);

un = sum(f);
u = zeros(un,1);
su = zeros(un,1);

% Create vector of units
for i = 1:n
    u(1+sum(f(1:i-1)):sum(f(1:i))) = i;
    su(1+sum(f(1:i-1)):sum(f(1:i))) = s(i);
end

aL1 = zeros(nk,length(unique(s)));
aPFE = zeros(un,length(unique(s)),nk);
iL1 = aL1;L2 = aL1;
iPFE = aPFE;aFV = aPFE;Cyc = aPFE;

% Execute NK repetitions of the DiMA analysis
for i = 1:nk
    [AM,IM] = getEMs(EGM,CM,u,su);
```

```
35      [aL1(i,:),aPFE(:,:,i),iL1(i,:),iPFE(:,:,i)] = getPFEdata(AM,IM);
36      [L2(i,:),aFV(:,:,i)] = getFVdata(AM);
37      Cyc(:,:,i) = getCYCdata(AM,s);
38      fprintf('  Case %i of %i completed\n',i,nk);
39  end
40
41  PFEn = aPFE - iPFE; % Calculate Net PFE
42
43  PFEprod=zeros(size(f),1);
44  PFEa=zeros(length(f),length(CM));PFEi=PFEa;
45  PFEnet=PFEa;FV=PFEa;CYCn=PFEa;
46
47  % Normalize Metrics and Compute Means
48  for i = 1:n
49      PFEa(i,:) = mean(mean(aPFE(u==i,:,:),3));
50      PFEi(i,:) = mean(mean(iPFE(u==i,:,:),3));
51      PFEprod(i) = sum(mean(prod(aPFE(u==i,:,:),2),3))/f(i);
52      PFEnet(i,:) = mean(mean(PFEn(u==i,:,:),3));
53      FV(i,:) = mean(mean(aFV(u==i,:,:),3),1);
54      CYCn(i,:) = sum(mean(Cyc(u==i,:,:),3));
55  end
56  out.CNEa = aL1 / un;
57  out.CNEi = iL1 / un;
58  out.L2   = L2;
59  out.FV   = FV;
60  out.PFEa = PFEa;
61  out.PFEi = PFEi;
62  out.PFEn = PFEnet;
63  out.PFEp = PFEprod;
64  out.Cyc  = CYCn;
65  varargout{1} = out;
66  fprintf(' Completed DiMA analysis\n');
67  end
68
69  function [AM,IM] = getEMs(EGM,CM,u,su)
```

```
70      un = length(u);
71      fn = size(EGM,3);
72      % Create Random Engagement Matrices for each Function
73      EM = rand(un,un,fn) < EGM(u,u,:);
74      for i = 1:fn
75          % Remove self-loops
76          EM(:,:,i) = EM(:,:,i) - diag(diag(EM(:,:,i)));
77      end
78
79      AM = zeros(un,un,length(CM));IM = AM;
80      % For every side
81      for s = 1:length(CM)
82          % For every function in the capability of that side
83          for i = 1:size(CM{s},1);
84              % Calculate the Active Engagement Matrix
85              AM(su == CM{s}(i,2),su == CM{s}(i,3),s) = AM(su == CM{s}(i,2),su ==
                  CM{s}(i,3),s) + EM(su == CM{s}(i,2),su == CM{s}(i,3),CM{s}(i,1));
86              if CM{i}(i,4)
87                  % And the Inert Engagement Matrix
88                  IM(su == CM{s}(i,2),su == CM{s}(i,3),s) = IM(su == CM{s}(i,2),su
                      == CM{s}(i,3),s) + EM(su == CM{s}(i,2),su == CM{s}(i,3),CM{s
                     }(i,1));
89              end
90          end
91      end
92      AM = double(AM > 0);
93      IM = double(IM > 0);
94  end
95
96  function [aL1,aPFE,iL1,iPFE] = getPFEdata(AM,IM)
97      n = size(AM,3);
98      aL1 = zeros(1,n);iL1=aL1;
99      aPFE = zeros(size(AM,1),n);iPFE = aPFE;
100     for i = 1:n
101         [aL1(1,i),aPFE(:,i)] = getPFE(AM(:,:,i));
```

```
102         [iL1(1,i),iPFE(:,i)] = getPFE(IM(:,:,i));
103     end
104 end
105 function [L2,FV] = getFVdata(AM)
106     n = size(AM,3);
107     L2 = zeros(1,n);
108     FV = zeros(size(AM,1),n);
109     for i = 1:n
110         [L2(i),FV(:,i)]  = getFV(AM(:,:,i));
111     end
112 end
113 function [CYC] = getCYCdata(AM,s)
114     CYC = zeros(size(AM,1),size(AM,3));
115     for i = 1:size(AM,3)
116         CYC(:,i) = getCyc(AM(:,:,i),sum(s==i)+1);
117     end
118 end
119
120 function [Lambda1,PFE] = getPFE(mat)
121 [V,D] = eig(mat');
122 D = diag(D);
123 j = find(imag(D) == 0);
124 [Lambda1,i] = max(D(j));
125 PFE = abs(V(:,j(i)));
126 end
127 function [D,V,n,unD] = getFV(net)
128 eps = 10^-12;
129 L = diag(sum(net)) - net';
130 [V,D] = eig(L');D = diag(D);
131 j = find(imag(D) == 0 & abs(D) > eps);
132 [ignore,i] = sort((D(j)),'ascend');
133 D = D(j(i(1)));
134 V = V(:,j(i(1)));
135 if nargout > 2
136     net = (net + net') > 0;
```

```
137     L = diag(sum(net)) - net';
138     [ignore,unD] = eig(L');unD = diag(unD);
139     n = sum(unD ≤ eps & imag(unD) == 0);
140     j = find(imag(unD) == 0 & abs(unD) > eps);
141     [ignore,i] = sort((unD(j)),'ascend');
142     unD = unD(j(i(1)));
143 end
144 end
145 function [cyc] = getCyc(net,lmax)
146 n = net ≠ 0;
147 for i = 1:lmax
148   n = (n ^ 2) > 0;
149 end
150 cyc = diag(n);
151 end
```

REFERENCES

[1] "architecture, $n.^{1,2b,5}$." Merriam-Webster Online. Merriam-Webster Online Dictionary. 2009. Accessed on 5 Jun. 2009. <http://www.merriam-webster.com/dictionary/architecture>.

[2] "architecture, $n.^{1,5,6}$." The Oxford English Dictionary. 2nd ed. 1989. OED Online. Oxford University Press. Accessed on 8 Jun. 2009 <http://dictionary.oed.com/cgi/entry/50011576>.

[3] "behavior, $n.^{5}$." The Oxford English Dictionary. 2nd ed. 1989. OED Online. Oxford University Press. Accessed on 12 Dec. 2007 <http://dictionary.oed.com/cgi/entry/50019665>.

[4] "model, $n.^{8.a}$." The Oxford English Dictionary. 2nd ed. 1989. OED Online. Oxford University Press. Accessed on 8 Dec. 2007 <http://dictionary.oed.com/cgi/entry/00313038>.

[5] "understand, $v.^{1.a,2,4.b}$." The Oxford English Dictionary. 2nd ed. 1989. OED Online. Oxford University Press. Accessed on 8 Jun. 2009 <http://dictionary.oed.com/cgi/entry/50265052>.

[6] "understanding, $vbl.n.^{1.b,1.c}$." The Oxford English Dictionary. 2nd ed. 1989. OED Online. Oxford University Press. Accessed on 8 Jun. 2009 <http://dictionary.oed.com/cgi/entry/50265058>.

[7] "Flexible analysis modeling and exercise system (flames)." Computer Program, Ternion, Inc., Huntsville, AL, December 2004.

[8] "System of systems systems engineering guide: Considerations for systems engineering in a system of systems environment," December 22 2006. Version 0.9.

[9] "Integrated deepwater system fact sheet," 1900 Half St. SW, Washington, D.C. 20593, September 2007.

[10] Abel, D. L. and Trevors, J. T., "Three subsets of sequence complexity and their relevance to biopolymeric information," *Theoretical Biology and Medical Modelling*, vol. 2, no. 1, pp. 29–45, 2005.

[11] AbuSharekh, A., Kansal, S., Zaidi, A. K., and Levis, A. H., "Modeling Time in DoDAF Compliant Executable Architectures," in *Proceedings of the 2007 Conference on Systems Engineering Research*, (Hoboken, NJ), March 2007.

[12] Ackoff, R. L., "From data to wisdom," *Journal of Applied Systems Analysis*, vol. 16, pp. 3–9, 1989.

[13] Ackoff, R. L., *Ackoff's Best: His Classic Writings on Management*. Wiley, 1999.

[14] Ackoff, R. L., ed., *Re-Creating the Corporation: A Design of Organizations for the 21st Century*. Oxford University Press, 1999.

[15] Adkins, G. and Pooch, U. W., "Computer simulation: a tutorial," *Computer*, vol. 10, pp. 12–17, April 1977.

[16] Albert, R. and Barabási, A.-L., "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, January 2002.

[17] Albert, R., Jeong, H., and Barabási, A.-L., "Error and attack tolerance of complex networks," *Nature*, vol. 406, pp. 378–382, July 2000.

[18] Alberts, D. S., *Information Age Transformation: Getting to a 21st Century Military*. Command and Control Research Program Publications Series, 2002.

[19] Alberts, D. S. and Czerwinski, T. J., eds., *Complexity, Global Politics, and National Security*. National Defense University, 1997.

[20] Alberts, D. S., Garstka, J. J., Hayes, R. E., and Signori, D. A., *Understanding Information Age Warfare*. Command and Control Research Program Publications Series, 2001.

[21] Alberts, D. S., Garstka, J. J., and Stein, F. P., *Network Centric Warfare: Developing and Leveraging Information Superiority*. Command and Control Research Program Publications Series, 2nd ed., 1999.

[22] Aldana, M., "Dynamics of boolean networks with scale-free topology," *Physica D: Nonlinear Phenomena*, vol. 185, pp. 45–66, October 2003.

[23] Alexander, R., "Models, gaming, and simulation: Session 1 - introduction," January 2009.

[24] Alexandrov, N. M., Kincaid, R. K., and Vargo, E. P., "Toward optimal transport networks," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. 2008-5814, AIAA, September 2008.

[25] Allsopp, D. N., Beautement, P., Kirton, M., Bradshaw, J. M., Suri, N., Tate, A., and Burstein, M., "The coalition agents experiment: Network enabled coalition operations," *Journal of Defence Science*, vol. 8, no. 3, pp. 130–141, 2003.

[26] Alspaugh, C., Hepner, T., Tran, C., Youm, W., Legaspi, A. K., Ferenci, S., Fujimoto, R., and Choi, M., "Efforts to Enhance Interoperability for NETWARS," *Simulation Series*, vol. 36, no. 4, p. 545, 2004.

[27] Amin, M., "National infrastructures as complex interactive networks," in *Automation, Control, and Complexity: An Integrated Approach* (Samad and Weyrauch, eds.), (3412 Hillview Avenue Palo Alto, CA 94304-1395, USA), pp. 263–286, Electric Power Research Institute, John Wiley and Sons, 2000.

[28] Amin, M., "Restructuring the electric enterprise, simulating the evolution of the electric power industry with intelligent adaptive agents," in *Market Analysis and Resource Management* (Faruqui, A. and Eakin, K., eds.), pp. 2–16, Electric Power Research Institute, Kluwer Publishers, March 2002.

[29] AND, J. G. K., "Apl integrated multi-warfare simulation (aims): Considering resource conflict resolution and warfare area inter-dependencies in multi-warfare analyses," in *73rd MORS Symposium*, June 2005.

[30] ANDERSON, P. W., "More is different: Broken symmetry and the nature of the hierarchical structure of science," *Science*, vol. 177, pp. 393–396, August 1972.

[31] ANDERSON, P., "Complexity theory and organization science," *Organization Science*, vol. 10, pp. 216–232, May-June 1999.

[32] ARGO, H., PRINCE, J., BROSS, P., JOHNSON, H., BLACKSTEN, H. R., BURDICK, C., GRAF, H., HUYNH, J., GRELL, M., and MACQUEEN, D., "Heterogeneous agent operations in jwars," May 2003. Presented at the 12th BRIMS Conference.

[33] ARTELLI, M. J. and DECKRO, R. F., "Modeling the lanchester laws with system dynamics," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 5, p. Online, January 2008.

[34] ATHANS, M., "Command and control (c2) theory: A challenge to control science," *IEEE Transactions on Automatic Control*, vol. 32, pp. 286–293, April 1987.

[35] ATKINSON, S. R. and MOFFAT, J., *The Agile Organization: From Informal Networks to Complex Effects and Agility*. Command and Control Research Program Publications Series, 2005.

[36] AZIZ-ALAOUI, M. and BERTELLE, C., eds., *Emergent Properties in Natural and Artificial Dynamical Systems*. Springer, 2006.

[37] BACHMAN, C. W., "The programmer as navigator," *Communications of the ACM*, vol. 16, no. 11, pp. 653–658, 1973.

[38] BALB, G. and CHIOLA, G., "Stochastic petri net simulation," in *Proceedings of the 1989 Simulation Winter Conference Proceedings* (MACNAIR, E., MUSSELMAN, K., and HEIDELBERGER, P., eds.), pp. 266–276, 1989.

[39] BANKES, S. C., "Tools and techniques for developing policies for complex and uncertain systems," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 7263–7266, May 2002.

[40] BAR-YAM, Y., *Dynamics of Complex Systems*. Perseus Books, 1997.

[41] BAR-YAM, Y., "Multiscale representation phase i," Tech. Rep. F30602-01-C-0030, New England Complex Systems Institute, August 2001.

[42] BAR-YAM, Y., "Large scale engineering and evolutionary change: Useful concepts for implementation of forcenet," Tech. Rep. F30602-02-C-0158, New England Complex Systems Institute, September 2002.

[43] BAR-YAM, Y., "Complexity of military conflict: Multiscale complex systems analysis of littoral warfare," Tech. Rep. F30602-02-C-0158, New England Complex Systems Institute, April 2003.

[44] BARABÁSI, A.-L., *Linked: How Everything is Connected to Everything Else and What it Means for Business, Science, and Everyday Life*. Basic Books, 2003.

[45] Barahona, F., Chowdhary, P., Ettl, M., Huang, P., Kimbrel, T., Ladanyi, L., Lee, Y., Schieber, B., Sourirajan, K., Sviridenko, M., and Swirszcz, G., "Inventory allocation and transportation scheduling for logistics of network-centric military opeartions," *IBM Journal of Research and Development*, vol. 51, pp. 391–408, March/June 2007.

[46] Barahona, M. and Pecora, L. M., "Synchronization in small-world systems," *Physical Review Letters*, vol. 89, p. 054101, July 2002.

[47] Bates, J. E. and Shepard, H. K., "Measuring complexity using information fluctuation," *Physics Letters A*, vol. 172, pp. 416–425, January 1993.

[48] Battilega, J. A. and Grange, J. K., eds., *The Military Applications of Modeling*. Wright-Patterson AFB, OH: Air Force Institute of Technology Press, 1984.

[49] Baumgarten, E. and Silverman, S. J., "Dynamic DoDAF and Executable Architectures," in *Proceedings from the MILCOM 2007 Military Communications Conference*, pp. 1–5, IEEE, October 2007.

[50] Bennett, C. H., "On the nature and origin of complexity in discrete, homogeneous, locally-interacting systems," *Foundations of Physics*, vol. 16, pp. 585–592, June 1986.

[51] Bentley, K., "The problems faced when defining emergence." History and Philosophy of Adaptive Systems EASy MSc COGS GRC University of Sussex, February 2002.

[52] Biest, A. V., Robert, F., Verkest, D., and Vernalde, S., "A taxonomy for technology extrapolation," in *In Proceedings of the 23rd NORCHIP Conference*, pp. 98–101, Nov. 2005.

[53] Bigelow, J. H. and Davis, P. K., "Implications for model validation of multiresolution, multiperspective modeling (mrmpm) and exploratory analysis," Monograph Reports MR-1750-AF, RAND Corporation, 2003.

[54] Biltgen, P. T., *A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems*. PhD thesis, Georgia Institute of Technology, May 2007.

[55] Blackwell, A. F. and Engelhardt, Y., "A taxonomy of diagram taxonomies," in *In Proceedings of Thinking with Diagrams 98: Is there a science of diagrams?*, pp. 60–70, 1998.

[56] Boccara, N., *Modeling Complex Systems*. Springer, 2004.

[57] Bonacich, P. F., "Power and centrality: A family of measures," *The American Journal of Sociology*, vol. 92, pp. 1170–1182, March 1987.

[58] Borowska, M., Oczeretko, E., Mazurek, A., Kitlas, A., and Kuć, P., "Application of the lempel-ziv complexity measure to the analysis of biosignals and medical images," *Roczniki Akademii Medycznej w Bialyamstonku*, vol. 50, no. 2, pp. 29–30, 2005.

[59] Borshchev, A. and Filippov, A., "From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools," in *The 22nd International Conference of the System Dynamics Society*, (Oxford, United Kingdom), 2004.

[60] Bounova, G. and de Weck, O., "Augmented network models for engineering system design," in *International Conference on Complex Systems 2006 (ICCS2006)*, (Boston, MA, USA), The New England Complex Systems Institute (NECSI), June 2006.

[61] Brewer, G. D. and Shubik, M., *The War Game: A Critique of Military Problem Solving*. Harvard University Press, 1979.

[62] Brinn, M., Berliner, J., Helsinger, A., Wright, T., Dyson, M., Rho, S., and Wells, D., "Extending the limits of DMAS survivability: the UltraLog project," *IEEE Intelligent Systems*, vol. 19, pp. 53–61, September/October 2004.

[63] Britt, R. R., "Universe 156 billion light- years wide," May 24 2004.

[64] Brooks, R. J., "Some thoughts on conceptual modelling: Performance, complexity and simplification," in *Proceedings of the 2006 OR Society Simulation Workshop*, 2006.

[65] Brooks, R. J. and Shi, D.-F., "The calibration of agent-based simulation models and their use for prediction," in *Proceedings of the 2006 OR Society Simulation Workshop*, 2006.

[66] Brown, F. T., *Engineering System Dynamics: A Unified Graph-Centered Approach*. CRC Press, 2nd ed., 2007.

[67] Buss, A. H. and Sánchez, P. J., "Simple movement and detection in discrete event simulation," in *Proceedings of the 2005 Winter Simulation Conference* (Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., eds.), pp. 992–1000, 2005.

[68] Calhoun, M. T., "Complexity and army transformation," in *Proceedings of the 2004 International Conference on Complex Systems*, 2004.

[69] Cares, J. R., "The use of agent-based models in military concept development," in *Proceedings of the 2002 Winter Simulation Conference*, pp. 935–939, 2002.

[70] Cares, J. R., "Rule sets for sense and respond logistics: The logic of demand networks," tech. rep., Alidade Incorporated, March 2004.

[71] Cares, J. R., "Distributed adaptive logistics," *Information Age Warfare Quarterly*, vol. 1, no. 1, pp. 4–12, 2005.

[72] Cares, J. R., *Distributed Networked Operations*. Newport, Rhode Island: Alidade Press, 1st ed., 2005.

[73] Cares, J. R., "Distributed networked operations and sense and respond logistics," January 2007.

[74] Carlson, J. M. and Doyle, J., "Complexity and robustness," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 2538–2545, February 2002.

[75] Cartier, M., "An agent-based model of innovation emergence in organizations: Renault and ford through the lens of evolutionism," *Computational & Mathematical Organization Theory*, vol. 10, pp. 147–153, July 2004.

[76] Castro e Silva, A., Kamphorst Leal da Silva, J., and Mendes, J. F. F., "Scale-free network with boolean dynamics as a function of connectivity," *Physical Review E*, vol. 70, p. 066140, December 2004.

[77] CDM Technologies, "Sea basing pilot assessment project," tech. rep., CDM Technologies, Inc., San Luis Obispo, CA, September 2006.

[78] Cebrowski, A. K., "Military transformation: A strategic approach." Office of Force Transformation, Fall 2003.

[79] Cebrowski, A. K., *The Implementation of Network-Centric Warfare*. Government Printing Office, 2005.

[80] Chaitin, G. J., "On the length of programs for computing finite binary sequences," *Journal of the Association for Computing Machinery*, vol. 13, no. 4, pp. 547–569, 1966.

[81] Champagne, L. E. and Hill, R. R., "Multi-agent simulation analysis: Bay of biscay case study," in *Proceedings of SimTecT 2003*, pp. 26–29, 2003.

[82] Champagne, L. E. and Hill, R. R., "Simulation validation with historic outcomes," in *Proceedings of the 2005 Winter Simulation Conference*, pp. 1138–1147, 2005.

[83] Chen, H.-M., "An inverse problem of the lanchester square law in estimating time-dependent attrition coefficients," *Operations Research*, vol. 50, pp. 389–394, March-April 2002.

[84] Chow, J. H. and Kokotovic, P. V., "Time scale modeling of sparse dynamic networks," *IEEE Transactions on Automatic Control*, vol. 30, pp. 714–722, August 1985.

[85] Claussen, J. C., "Offdiagonal complexity: A computationally quick complexity measure for graphs and networks," *Physica A: Statistical Mechanics and its Applications*, vol. 375, no. 1, pp. 365–373, 2007.

[86] Corning, P. A., "Complexity is just a word!," *Technological Forecasting and Social Change*, vol. 59, pp. 197–200, 1998.

[87] Cotsaftis, M., "What makes a system complex? an approach to self-organization and emergence." Online, June 2007.

[88] Couture, M., "Complexity and chaos - state-of-the-art; formulations and measures of complexity," Technical Note TN 2006-451, Defence R&D Canada - Valcartier, September 2007.

[89] Couture, M., "Complexity and chaos - state-of-the-art; overview of theoretical concepts," Technical Memorandum TM 2006-453, Defence R&D Canada - Valcartier, August 2007.

[90] Crutchfield, J. P. and Young, K., "Inferring statistical complexity," *Physical Review Letters*, vol. 63, no. 2, pp. 105–108, 1989.

[91] Davis, P. K., "Effects-based operations: A grand challenge for the analytical community," Monograph MR-1477-USJFCOM/AF, RAND Corporation, 2001.

[92] Davis, P. K. and Bigelow, J. H., "Experiments in multiresolution modeling (mrm)," Monograph MR-1004-DARPA, RAND Corporation, 1998.

[93] Davis, P. K. and Bigelow, J. H., "Motivated metamodels: Synthesis of cause-effect reasoning and statistical metamodeling," Monograph MR-1570, RAND Corporation, 2003.

[94] Davis, R. and Kearns, M., "Simulations role in the defence experimental framework," in *Proceedings of SimTecT 2002*, pp. 139–144, Simulation Industry Association of Australia, 2002.

[95] Dawkins, R., *The Blind Watchmaker*. Norton & Company, Inc, 1986.

[96] Deguet, J., Demazeau, Y., and Magnin, L., "Elements about the emergence issue: A survey of emergence definitions," *Complexus*, vol. 3, pp. 24–31, 2006.

[97] Dekker, A. H. and Colbert, B. D., "Network robustness and graph topology," in *ACSC '04: Proceedings of the 27th Australasian conference on Computer science*, vol. 26, (Darlinghurst, Australia, Australia), pp. 359–368, Australian Computer Society, Inc., 2004.

[98] Department of Defense, *DoD Modeling and Simulation (M&S) Glossary*, December 1997.

[99] Department of the Army, Commander, United States Army John F. Kennedy Special Warfare Center and School, ATTN: AOJK-DT-SFD, Fort Bragg, North Carolina 28310-5000., *Special Forces Use of Pack Animals*, June 2004. FM 3-05.213 (FM 31-27).

[100] Derrida, B. and Pomeau, Y., "Random networks of automata: A simple annealed approximation," *EPL (Europhysics Letters)*, vol. 1, no. 2, pp. 45–49, 1986.

[101] Derrida, B. and Stauffer, D., "Phase transitions in two-dimensional kauffman cellular automata," *Europhysics Letters*, vol. 2, pp. 739–745, November 1986.

[102] DeStefano, G. V., "Agent Based Simulation SEAS Evaluation of DoDAF Architecture," Master's thesis, Air Force Institute of Technology, March 2004.

[103] Dewdney, A. K., "Computer recreations: The hodge-podge machine makes waves," *Scientific American*, vol. 43, pp. 104–107, 1988.

[104] Díaz, J., Petit, J., and Serna, M., "A survey of graph layout problems," *ACM Computing Surveys*, vol. 34, pp. 313–356, September 2002.

[105] Dieter, G. E., *Engineering Design: A Materials and Processing Approach*. McGraw-Hill, 3rd ed., 2000.

[106] Dobson, K. G., "Complexity science will transform logistics," *U.S. Naval Institute Proceedings*, vol. 130, no. 4, pp. 74–76, 2004.

[107] DoDAF Working Group, *DoD Architecture Framework Version 1.5*, vol. II: Product Descriptions. United States Department of Defense, April 2007.

[108] Dodziuk, J. and Kendall, W. S., *From Local Times to Global Geometry*, vol. 150, ch. Combinatorial Laplacians and Isoperimetric Inequality, pp. 68–74. Pitman Research Notes in Mathematics Series, 1985.

[109] Dorogovtsev, S. N. and Mendes, J. F. F., "Evolution of networks," *Advances in Physics*, vol. 51, pp. 1079–1187, June 2002.

[110] Downey, A. B., *Computational Modeling and Complexity Science*. Green Tea Press, 2008.

[111] Doyle, J. K. and Ford, D. N., "Mental models concepts for system dynamics research," *System Dynamics Review*, vol. 14, no. 1, pp. 3–29, 1998.

[112] Doyle, J. K. and Ford, D. N., "Mental models concepts revisited: some clarifications and a reply to lane," *System Dynamics Review*, vol. 15, no. 4, pp. 411–415, 1999.

[113] Dueñas Osorio, L. A., *Interdependent Response of Networked Systems to Natural Hazards and Intentional Disruptions*. PhD thesis, Georgia Institute of Technology, December 2005.

[114] Duke, C. B., Hopcroft, J. E., Arkin, A. P., Armstrong, R. E., Barabási, A.-L., Brachman, R. J., Broome, N. L., DeMillo, S. D. R. A., Hilsman, W. J., Leland, W. E., Malone, T. W., Murray, R. M., Pellicci, J., Silver, P. A., Riper, P. K. V., and Watts, D. J., *Network Science*. National Academies Press, 2005.

[115] Dupuy, T. N., *Numbers, Predictions and War: Using History to Evaluate Combat Factors and Predict the Outcome of Battles*. Macdonald and Jane's, 1st ed., 1979.

[116] Dupuy, T. N., *Numbers, Predictions and War: Using History to Evaluate Combat Factors and Predict the Outcome of Battles*. Hero Books, 2nd ed., 1985.

[117] Dupuy, T. N., *Attrition: Forecasting Battle Casualties and Equipment Losses in Modern War*. Fairfax, VA: NOVA Publications, 1990.

[118] Eberlein, R. L., "Simplification and understanding of models," *System Dynamics Review*, vol. 5, no. 1, pp. 51–68, 1989.

[119] Edmonds, B., "What is complexity? - the philosophy of complexity per se with application to some examples in evolution," in *The Evolution of Complexity* (Heylighen, F. and Aerts, D., eds.), Kluwer, Dordrecht, 1999.

[120] Ender, T. R., *A Top-Down, Hierarchical, System-of-Systems Approach to the Design of an Air Defense Weapon*. PhD thesis, Georgia Institute of Technology, August 2006.

[121] Fagiolo, G., Moneta, A., and Windrum, P., "A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems," *Computational Economics*, vol. 30, p. 195226, September 2007.

[122] Fax, J. A. and Murray, R. M., "Graph laplacians and stabilization of vehicle formations," in *15th IFAC World Congress*, 2002.

[123] Fax, J. A. and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1465–1476, September 2004.

[124] Ferenci, S. L., Choi, M., Evans, J., Fujimoto, R. M., Alspaugh, C., and Legaspi, A. K., "Experiences integrating NETWARS with the naval simulation system using the high level architecture," in *IEEE 2004 Military Communications Conference*, vol. 3, pp. 1395–1401, 2004.

[125] Ferguson, A., *SAS: Special Air Service*. Rosen Publishing Group, March 2003.

[126] Ferguson, R., "Modelling social systems," August 2006.

[127] Fiddaman, T. S., "Exploring policy options with a behavioral climateeconomy model," *System Dynamics Review*, vol. 18, pp. 243–267, 2002.

[128] Fishwick, P. A., "An integrated approach to system modeling using a synthesis of artificial intelligence, software engineering and simulation methodologies," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 2, pp. 307–330, October 1992.

[129] Fishwick, P. A., "A taxonomy for simulation modeling based on programming language principles," *IIE Transactions*, vol. 30, pp. 811–820, September 1998.

[130] Fishwick, P. A. and Miller, J. A., "Ontologies for modeling and simulation: Issues and approaches," in *Proceedings of the 2004 Winter Simulation Conference* (Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., eds.), 2004.

[131] Forrester, J. W., "System dynamics, systems thinking, and soft OR," *System Dynamics Review*, vol. 10, no. 2, pp. 245–259, 1994.

[132] Forrester, J. W., *Industrial Dynamics*. Cambridge, MA: M.I.T. Press, 1961. ISBN 0262560011.

[133] Forrester, J. W., "Designing the future," in *Speech given at the Universidad de Sevilla*, December 15 1998.

[134] Forsythe, S. L., North, P. D., and Barnes, V. B., "Evaluation of net-centric command and control via a multi-resolution modeling evaluation framework," in *10th International Command and Control Research and Technology Symposium: The Future of C2*, June 2005.

[135] Fossett, C. A., Harrison, D., Weintrob, H., and Gass, S. I., "An assessment procedure for simulation models: A case study," *Operations Research*, vol. 39, pp. 710–723, September-October 1991.

[136] Fox, D. B., *New Challenges, New Tools for Defense Decisionmaking*, ch. Chapter 10: Using Exploratory Modeling, pp. 285–298. No. MR1576, RAND Corporation, 2003.

[137] Fox, D. B. and Jones, C. M., *JICM 3.0: Documentation and Tutorial*. RAND Corporation, 1998.

[138] Franklin, J. E., Carmody, C. L., Keller, K., Levitt, T. S., and Buteau, B. L., "Expert system technology for the military: Selected samples," in *Proceedings of the IEEE* [179], pp. 1327–1366.

[139] Franklin, S. and Graesser, A., "Is it an agent, or just a program?: A taxonomy for autonomous agents," in *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996.

[140] Fraundorf, P., "A simplex model for layered niche networks," *Complexity*, vol. 13, no. 6, pp. 29–39, 2008.

[141] Frigg, R. and Hartmann, S., "Models in science." Stanford Encyclopedia of Philosophy, February 2006. Accessed online November 2007. <http://plato.stanford.edu/entries/models-science/>.

[142] Frits, A. P., *Formulation of an Integrated Robust Design and Tactics Optimization Process for Undersea Weapon Systems*. PhD thesis, Georgia Institute of Technology, December 2004.

[143] Ganapathy, S. and Hill, R. R., "Dynamic path-planning for search and destroy missions - the bay of biscay scenario," in *Proceedings of the 35th conference on Winter simulation: driving innovation*, pp. 999–1003, 2003.

[144] Garcia, J., "Executable architecture analysis modeling," in *Proceedings of the 2007 Spring Simulation Multiconference*, vol. 3, pp. 177–184, Society for Computer Simulation International, 2007.

[145] Garifullin, M., Borshchev, A., and Popkov, T., "Using anylogic and agent-based approach to model consumer market," in *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, 9-13 September 2007.

[146] Gell-Mann, M., *The Quark and the Jaguar: Adventures in the Simple and the Complex*. Owl Books, 1995. ISBN 0805072535.

[147] Gell-Mann, M., "What is complexity?," *Complexity*, vol. 1, no. 1, pp. 16–19, 1995.

[148] Gell-Mann, M. and Lloyd, S., "Information measures, effective complexity, and total information," *Complexity*, vol. 2, no. 1, pp. 44–52, 1996.

[149] Genrich, H. J. and Lautenbach, K., "System modelling with high-level petri nets," *Theoretical Computer Science*, vol. 13, no. 1, pp. 109–135, 1981.

[150] Golfarelli, M., Rizzi, S., and Proli, A., "Designing what-if analysis: Towards a methodology," in *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP*, 2006.

[151] Gonzales, D., Moore, L. R., Pernin, C. G., Matonick, D. M., and Dreyer, P., "Assessing the value of information superiority for ground forces-proof of concept," tech. rep., Rand Corporation, 2001.

[152] Gordon, G., "Simulation languages for discrete systems," in *Proceedings IBM Scientific Computing Symposium on Simulation Models and Gaming*, (White Plains, New York), pp. 101–118, 1966.

[153] Gordon, G., *System Simulation.* Prentice Hall, 1969.

[154] Griendling, K., *Architecture and Principles of Systems Engineering*, ch. 11: DoDAF and MODAF Artifacts, p. To Be Published. Auerbach Publications, 1st ed., July 2009.

[155] Grössler, A., Milling, P., and Winch, G., "Perspectives on rationality in system dynamics - a workshop report and open research questions," *System Dynamics Review*, vol. 20, no. 1, pp. 75–87, 2004.

[156] Grynkewich, A. and Reifel, C., "Modeling jihad: A system dynamics model of the salafist group for preaching and combat financial subsystem," *Strategic Insights*, vol. V, p. Online, November 2006.

[157] Gustafsson, L., "Poisson simulation - a method for generating stochastic variations in continuous system simulation," *Simulation*, vol. 74, pp. 264–274, May 2000.

[158] Gustafsson, L. and Sternad, M., "Bringing consistency to simulation of population models poisson simulation as a bridge between micro and macro simulation," *Mathematical Biosciences*, vol. 209, pp. 361–385, 2007.

[159] Habeck, M., *Storm of Steel: The Development of Armor Doctrine in Germany and the Soviet Union, 1919-1939.* Cornell University Press, 2003.

[160] Hamber, B., "T.loads abbreviated systems architecture," in *Proceedings of the 2001 Winter Simulation Conference* (Peters, B. A., Smith, J. S., Medeiros, D. J., and Rohrer, M. W., eds.), pp. 749–757, 2001.

[161] Hartley III, D. S., *Topics on Operations Research: Predicting Combat Effects.* INFORMS, 2001.

[162] Hartley III, D., "A Mathematical Model of Attrition Data," *Naval Research Logistics*, vol. 42, pp. 585–607, June 1995.

[163] Hastings, W. K., "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[164] He, C., Pan, Y., Shi, P., Li, X., Chen, J., Li, Y., and Li, J., "Developing land use scenario dynamics model by the integration of system dynamics model and cellular automata model," in *Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2647–2650, September 2004.

[165] Helbing, D., Farkas, I., and Vicsek, T., "Simulating dynamical features of escape panic," *Nature*, vol. 407, pp. 487–490, 28 September 2000.

[166] Helmbold, R. L., "A "universal" attrition model," *Operations Research*, vol. 14, pp. 624–635, July-August 1966.

[167] Ho, J. H., "Waging effects-based operations," *Security Challenges*, vol. 2, no. 1, pp. 157–168, 2006.

[168] Holland, J. H., *Hidden Order: How Adaptation Builds Complexity.* Reading, Massachusetts: Helix Books, 1995.

[169] Holland, J. H., *Emergence: From Chaos to Order*. Reading, Massachusetts: Helix Books, 1st ed., 1998.

[170] Hughes, Wayne P., J., ed., *Military Modeling for Decision Making*. Military Operations Research Society, 3rd ed., 1997.

[171] Ilachinski, A., "Land warfare and complexity, part i: Mathematical backgorund and technical sourcebook," Memorandum CIM 461, Center for Naval Analyses, July 1996.

[172] Ilachinski, A., "Land warfare and complexity, part ii: An assessment of the applicability of nonlinear dynamics and complex systems theory to the study of land warfare," Memorandum CRM 96-68, Center for Naval Analyses, July 1996.

[173] Ilachinski, A., "Towards a science of experimental complexity: An artificial-life approach to modeling warfare," in *Proceedings of 5th Experimental Chaos Conference*, 1999.

[174] Infield, G. B., *Skorzeny: Hitler's commando*. Hippocrene Books, 1981.

[175] Jain, S. and Krishna, S., "Autocatalytic sets and the growth of complexity in an evolutionary model," *Physical Review Letters*, vol. 81, pp. 5684–5687, December 1998.

[176] Jain, S. and Krishna, S., "Emergence and growth of complex networks in adaptive systems," *Computer Physics Communications*, vol. 121122, p. 116121, 1999.

[177] Jain, S. and Krishna, S., *Handbook of Graphs and Networks: From the Genome to the Internet*, ch. 16. Graph Theory and the Evolution of Autocatalytic Networks, pp. 355–401. John Wiley and VCH publishers, 2002.

[178] Jain, S. and Krishna, S., "Large extinctions in an evolutionary model: The role of innovation and keystone species," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 2055–2060, February 2002.

[179] Jaiswal, N. K., *Military Operations Research: Quantitative Decision Making*. Springer, 1997.

[180] Jarvis, D. A., "Command and control network analysis," tech. rep., Alidade Incorporated, September 2004. Completed under Contract N66604-04-C-4596.

[181] Jensen, H. J., *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*. Cambridge University Press, January 1998.

[182] Johnson, S., *Emergence: the connected lives of ants, brains, cities and software*. Simon & Schuster, 2001.

[183] Johnson, S. E., Libicki, M. C., and Treverton, G. F., eds., *New Challenges, New Tools for Defense Decisionmaking*. RAND Corporation, 2003. Monograph Report MR-1576-RC.

[184] Kant, I., *Prolegomena to Any Future Metaphysics*. Liberal Arts Press (1950), 1783.

[185] Karanta, I., "Quantifying combat effects with spreadsheets," in *Finnish Operations Research Society*, 2008.

[186] Kaspar, F. and Schuster, H. G., "Easily calculable measure for the complexity of spatiotemporal patterns," *Physical Review A*, vol. 36, pp. 842–848, July 15 1987.

[187] Kauffman, S. A., "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, pp. 437–467, March 1969.

[188] Kauffman, S. A., *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[189] Kauffman, S. A., *At Home in the Universe: The Search for Laws of Self-organization and Complexity*. Oxford University Press, 1995.

[190] Kaufman, V. and Drossel, B., "Relevant components in critical random Boolean networks," *New Journal of Physics*, vol. 8, p. 228, 2006.

[191] Kincaid, R. K., Alexandrov, N., and Holroyd, M. J., "An investigation of synchrony in transport networks," *Complexity*, vol. 14, no. 4, pp. 34–43, 2008.

[192] Kiviat, P. J., "Digital computer simulation: Modeling concepts," Memo RM-5378-PR, RAND Corporation, August 1967.

[193] Kiviat, P. J., "Digital computer simulation: Computer programming languages," Tech. Rep. RM-5883-PR, RAND Corporation, January 1969.

[194] Kleijnen, J. P. C., Sanchez, S. M., Lucas, T. W., and Cioppa, T. M., "A user's guide to the brave new world of designing simulation experiments," Discussion Paper 200301, Tilburg University - CentER, January 2003. ISSN 0924-7815.

[195] Kleijnen, J. P. C., Sanchez, S. M., Lucas, T. W., and Cioppa, T. M., "State-of-the-art review: A users guide to the brave new world of designing simulation experiments," *INFORMS Journal on Computing*, vol. 17, pp. 263–289, Summer 2005.

[196] Klein, D. and Ivanciuc, O., "Graph cyclicity, excess conductance, and resistance deficit," *Journal of Mathematical Chemistry*, vol. 30, no. 3, pp. 271–287, 2001.

[197] Klügl, F., Oechslein, C., Puppe, F., and Dornhaus, A., "Multi-agent modeling in comparison to standard modelling," in *AIS 2002: Proceedings of AI, Simulation and Planning in High Autonomy Systems* (Barros, F. J. and Giambiasi, N., eds.), pp. 105–110, SCS Publishing House, 2002.

[198] Koch, C. and Laurent, G., "Complexity and the nervous system," *Science*, vol. 284, pp. 96–98, April 1999.

[199] Koehler, M., *Maneuver Warfare Science 2002*, ch. Emergence and the Military, Is It Important or Strictly for the Boids?, pp. 149–179. USMC Project Albert, June 2002.

[200] Koopman, J. S., Jacquez, G., and Chick, S. E., "New data and tools for integrating discrete and continuous population modeling strategies," *Annals of the New York Academy of Sciences*, vol. 954, pp. 268–294, December 2001.

[201] Kovalchik, J. and Labin, M. J. W., "Apl integrated multi-warfare simulation (aims): Providing resource conflict resolution in multi-warfare analyses," *Simulation Technology Magazine*, vol. 9, p. Online, September 2006.

[202] Kuhn, T., *The Structure of Scientific Revolutions*. University of Chicago Press, 1962.

[203] Kurzweil, R., "The law of accelerating returns." Published on KurzweilAI.net March 7, 2001, March 2001.

[204] Lanchester, F. W., *Aircraft in Warfare: The Dawn of the Fourth Arm*. Constable, 1916.

[205] Landauer, C., "Data, information, knowledge, understanding: Computing up the meaning hierarchy," in *1998 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2255–2260, October 1998.

[206] Latora, V. and Marchiori, M., "Efficient behavior of small-world networks," *Physical Review Letters*, vol. 87, p. 198701, November 2001.

[207] Latora, V. and Marchiori, M., "Is the boston subway a small-world network?," *Physica A*, vol. 314, pp. 109–113, 2002.

[208] Lauren, M. K. and Stephen, R. T., "Fractals and combat modeling: Using MANA to explore the role of entropy in complexity science," *Fractals*, vol. 10, no. 4, pp. 481–489, 2002.

[209] Lempel, A. and Ziv, J., "On the complexity of finite sequences," *IEEE Transactions on Information Theory*, vol. 22, pp. 75–81, January 1976.

[210] Lempert, R. J., "A new decision sciences for complex systems," *Proceedings of the National Academy of Sciences*, vol. 99, p. 73097313, May 2002.

[211] Levis, A. H., "Modeling and simulation for architecture assessment," April 20 2004.

[212] Levis, A. H. and Wagenhals, L. W., "C4ISR architectures: I. Developing a process for C4ISR architecture design," *Systems Engineering*, vol. 3, pp. 225–247, November 2000.

[213] Lewe, J.-H., *An Integrated Decision-Making Framework for Transportation Architectures: Application to Aviation Systems Design*. PhD thesis, Georgia Institute of Technology, April 2005.

[214] Li, M. and Vitányi, P., *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1997.

[215] Libicki, M. C., Gompert, D. C., Frelinger, D. R., and Smith, R., "Byting back: Regaining information superiority against 21st-century insurgents," Tech. Rep. MG-595/1-OSD, RAND Corporation, 2007.

[216] Lind, W. S., Schmitt, J. F., Sutton, J. W., and Wilson, G. I., "The changing face of war: Into the fourth generation," *Marine Corps Gazette*, vol. 73, pp. 22–26, October 1989.

[217] Lloyd, M., "Commanding agile mission groups: A speculative model," *Journal of Defence Science*, vol. 8, pp. 152–159, 2003.

[218] Lloyd, S. and Pagels, H., "Complexity as thermodynamic depth," *Annals of Physics*, vol. 188, pp. 186–213, November 1988.

[219] Lodwick, J., *Raiders from the Sea: The Story of the Special Boat Service in WWII.* Naval Institute Press, 1990.

[220] Lucas, T. W., "The stochastic versus deterministic argument for combat simulations: Tales of when the average won't do," *Military Operations Research*, vol. 5, no. 3, pp. 9–28, 2000.

[221] Lucas, T. W. and McGunnigle, J. E., "When is model complexity too much? illustrating the benefits of simple models with hughes' salvo equations," *Naval Research Logistics*, vol. 50, pp. 197 – 217, April 2003.

[222] Maani, K. E. and Maharaj, V., "Links between systems thinking and complex decision making," *System Dynamics Review*, vol. 20, no. 1, pp. 21–48, 2004.

[223] Machta, B. and Machta, J., "Parallel dynamics and computational complexity of network growth models," *Physical Review E*, vol. 71, p. 026704, February 2005.

[224] MacKerrow, E. P., "Understanding why dissecting radical islamist terrorism with agent-based simulation," *Los Alamos Science*, vol. 28, pp. 184–191, 2003.

[225] Maguire, P., "Using thunder for campaign studies," Tech. Rep. DSTO-TN-0303, DSTO Aeronautical and Maritime Research Laboratory: Air Operations Division, 2000.

[226] Maier, M. W. and Rechtin, E., *The Art of Systems Architecting.* CRC Press LLC, 2nd ed., 1997.

[227] Maier, M. W. and Rechtin, E., *The Art of Systems Architecting.* CRC Press, 2nd ed., 2000.

[228] Malamud, B. D. and Turcotte, D. L., "Cellular-automata models applied to natural hazards," *Computing in Science & Engineering*, vol. 2, pp. 42–51, May-June 2000.

[229] Martin-Löf, P., "The definition of random sequences," *Information and Control*, vol. 9, no. 6, pp. 602–619, 1966.

[230] Matsumura, J., *Exploring Advanced Technologies for the Future Combat Systems Program.* Rand Corporation, 2002.

[231] Maynard Smith, J., *The Problems of Biology.* Oxford University Press, 1986.

[232] McDonald, R. A., *Error Propagation and Metamodeling for a Fidelity Tradeoff Capability in Complex Systems Design.* PhD thesis, Georgia Institute of Technology, 2006.

[233] McHaney, R., *Computer Simulation: A Practical Perspective.* Academic Press, 1991.

[234] McIntyre, G. A., "Joint warfare system (jwars) output analysis," in *MORS Workshop: Operations Analysis Support to Network Centric Operations*, January 2004.

[235] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, June 1953.

[236] Meyer-Ortmanns, H., "Functional complexity measure for networks," *Physica A: Statistical and Theoretical Physics*, vol. 337, pp. 679–690, June 2004.

[237] Miller, J. A., Baramidze, G. T., Sheth, A. P., and Fishwick, P. A., "Investigating ontologies for simulation modeling," in *Proceedings of the 37th annual symposium on Simulation*, (Washington, DC), IEEE Computer Society, 2004.

[238] Mitchell, M., Hraber, P. T., and Crutchfield, J. P., "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *Complex Systems*, vol. 7, pp. 89–130, 1993.

[239] Mittal, S., "Extending DoDAF to Allow Integrated DEVS-Based Modeling and Simulation," *Journal of Defense Modeling and Simulation*, vol. 3, pp. 95–123, April 2006.

[240] Mittal, S., Mak, E., and Nutaro, J. J., "DEVS-Based Dynamic Model Reconfiguration and Simulation Control in the Enhanced DoDAF Design Process," *Journal of Defense Modeling and Simulation*, vol. 3, pp. 239–267, October 2006.

[241] Moffat, J., *Complexity Theory and Network Centric Warfare*. Command and Control Research Program Publications Series, 2003.

[242] Moffat, J. and Passman, M., "Metamodels and emergent behaviour in models of conflict," *Simulation Modelling Practice and Theory*, vol. 12, pp. 579–590, 2004.

[243] Mojtahedzadeh, M., Andersen, D., and Richardson, G. P., "Using digest to implement the pathway participation method for detecting influential system structure," *System Dynamics Review*, vol. 20, no. 1, pp. 1–20, 2004.

[244] Mordente, P., Needham, P., and Ogren, T. P., "Logistics for the 21st century: Deployment distribution operations center, quick fix or long-term solution?," *Air Force Journal of Logistics*, vol. 30, no. 4, pp. 82–92, 2006.

[245] Mortimer, G., *Stirling's Men: The Inside History of the SAS in World War II*. Weidenfeld & Nicolson, 2004.

[246] Motter, A. E., "Bounding network spectra for network design," *New Journal of Physics*, vol. 9, pp. 182–199, June 2007.

[247] Murphy, G., Shippy, D. J., and Luo, H. L., *Engineering Analogies*. Iowa State University Press, 1963.

[248] Nance, R. E., "A history of discrete event simulation programming languages," *ACM SIGPLAN Notices*, vol. 28, pp. 149 – 175, March 1993.

[249] Neelamkavil, F., *Computer Simulation and Modelling*. Wiley, January 1987.

[250] Nelson, R. J., "Structure of complex systems," in *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, vol. 2, pp. 523–542, 1976.

[251] Newman, M. E. J., "Assortative mixing in networks," *Physical Review Letters*, vol. 89, p. 208701, November 2002.

[252] Newman, M. E. J., "Random graphs as models of networks," tech. rep., Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM, 2002.

[253] Newman, M. E. J., "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[254] Nilsson, F., "Simplicity vs. complexity in the logistics discipline - a paradigm discourse," in *16th annual NOFOMA Conference*, 2004.

[255] Nishikawa, T. and Motter, A. E., "Maximum performance at minimum cost in network synchronization," *Physica D: Nonlinear Phenomena*, vol. 224, pp. 77–89, December 2006.

[256] O'Bryon, J. F., "Meet "master" modeling & simulation test & evaluation reform: Energizing the m&s support structure," *Program Manager (PM) Magazine*, vol. 6, pp. 29–34, March/April 1999.

[257] Oden, J. T., Belytschko, T., Fish, J., Hughes, T. J. R., Johnson, C., Keyes, D., Laub, A., Petzold, L., Srolovitz, D., and Yip, S., "Simulation-based engineering science: Revolutionizing engineering science through simulation," tech. rep., National Science Foundation, May 2006. Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science.

[258] Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520–1533, September 2004.

[259] Oliva, R., "Model structure analysis through graph theory: Partition heuristics and feedback structure decomposition," *System Dynamics Review*, vol. 20, no. 4, pp. 313–336, 2004.

[260] Ong, J. and Ling, M. F., "Using the joint integrated contingency model for campaign analysis," Technical Report DSTO-TR-1307, Defence Systems Analysis Division, May 2002.

[261] Osmundson, J. S., Gottfried, R., Kum, C. Y., Boon, L. H., Lian, L. W., Patrick, P. S. W., and Thye, T. C., "Process modeling: A systems engineering tool for analyzing complex systems," *Systems Engineering*, vol. 7, p. 320337, June 2004.

[262] Parunak, H. V. D. and Brueckner, S., "Modeling uncertain domains with polyagents," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, (New York, NY, USA), pp. 111–113, ACM, 2006.

[263] Pati, S., "The third smallest eigenvalue of the laplacian matrix," *The Electronic Journal of Linear Algebra*, vol. 8, pp. 128–139, August 2001.

[264] Perla, P. P., *The Art of Wargaming: A Guide for Professionals and Hobbyists*. Naval Institute Press, 1990.

[265] Petri, C. A., *Kommunikation mit Automaten.* PhD thesis, Institut fur Instrumentelle Mathematik, 1962.

[266] Poole, B. H., *A Methodology for the Robustness-Based Evaluation of Systems-of-Systems Alternatives using Regret Analysis.* PhD thesis, Georgia Institute of Technology, 2008.

[267] Popescu, C., Lobov, A., and Martinez Lastra, J. L., "An approach to computing all cycles in a graph," in *Proceedings from the 2006 IEEE International Conference on Industrial Informatics*, pp. 1273–1278, 2006.

[268] Popper, K. R., *The Logic of Scientific Discovery.* Routledge, 1959.

[269] Popper, K. R., *Conjectures and Refutations: The Growth of Scientific Knowledge.* Harper Torchbooks, 1963.

[270] Przemieniecki, J. S., *Mathematical methods in defense analyses.* Washington, D.C.: American Institute of Aeronautics and Astronautics, 2nd ed., 1994.

[271] Qudrat-Ullah, H., "Structural validation of system dynamics and agent-based simulation models," in *Proceedings 19th European Conference on Modelling and Simulation*, 2005.

[272] Ragsdell, G. and Wilby, J., eds., *Understanding Complexity.* New York: Springer, 2001.

[273] Rainey, L. B., ed., *Space Modeling And Simulation: Roles And Applications Throughout The System Life Cycle.* AIAA, 2004.

[274] Revinskiy, V. V., "Command and control system in the tactical echelon," *Military Thought*, vol. 13, no. 1, pp. 172–178, 2004.

[275] Richards, C. W., *A Swift, Elusive Sword: What If Sun Tzu and John Boyd Did a National Defense Review?* Center for Defense Information, 2001.

[276] Ring, S. J., Nicholson, D., Thilenius, J., and Harris, S., "An Activity-Based Methodology for Development and Analysis of Integrated DoD Architectures," in *In Proceedings of the 2004 Command and Control Research and Technology Symposium: The Power of Information Age Concepts and Technologies*, 2004.

[277] Ripley, R. F. and Larkin, G. R., "Applying complexity theory to aviation management," in *Proceedings of the AIAA 5th Aviation, Technology, Integration, and Operations (ATIO) Conference*, 26-28 September 2005.

[278] Robertson, J. I., "A method of computing survival probabilities of several targets versus several weapons," *Operations Research*, vol. 4, pp. 546–557, October 1956.

[279] Robinson, S., "Issues in conceptual modelling for simulation: Setting a research agenda," in *Proceedings of the 2006 OR Society Simulation Workshop*, 2006.

[280] Rolleston, M., Pernot, J., and Keeports, T., *U.S. Air Force Transformation Flight Plan.* U.S. Air Force, 2004.

[281] Rothenberg, J., *AI, Simulation & Modeling*, ch. The Nature of Modeling, pp. 75–92. John Wiley & Sons, Inc., August 1989.

[282] Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[283] Sakellariou, I., Kefalas, P., and Stamatopoulou, I., *Artificial Intelligence: Theories, Models and Applications*, vol. 5138, ch. Enhancing NetLogo to Simulate BDI Communicating Agents, pp. 263–275. Springer Berlin / Heidelberg, 2008.

[284] Santos, E., Zhao, Q., Pratto, F., Pearson, A. R., McQueary, B., Breeden, A., and Krause, L., "Modeling multiple communities of interest for interactive simulation and gaming: The dynamic adversarial gaming algorithm project," 2005.

[285] Saysel, A. K. and Barlas, Y., "Model simplification and validation with indirect structure validity tests," *System Dynamics Review*, vol. 22, no. 3, pp. 241–262, 2006.

[286] Schaffer, M. B., "A model of 21st century counterinsurgency warfare," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 4, p. Online, July 2007.

[287] Schaffernicht, M., "Detecting and monitoring change in models," *System Dynamics Review*, vol. 22, no. 1, pp. 73–88, 2006.

[288] Schieritz, N., "Integrating system dynamics and agent-based modeling," in *Proceedings of the 20th International Conference of the Systems Dynamics Society*, (Palermo), 2002.

[289] Schieritz, N. and Milling, P. M., "Modeling the forest or modeling the trees: A comparison of system dynamics and agent-based simulation," in *Proceedings of the 21st International Conference of the Systems Dynamics Society*, 2003.

[290] Scholl, H. J., "Agent-based and system dynamics modeling: A call for cross study and joint research," in *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.

[291] Schuster, P., "A beginning of the end of the holism versus reductionism debate?," *Complexity*, vol. 13, pp. 10–13, October 2007.

[292] Serra, R., Villani, M., and Agostini, L., "On the dynamics of random boolean networks with scale-free outgoing connections," *Physica A: Statistical Mechanics and its Applications*, vol. 339, pp. 665–673, August 2004.

[293] Shalizi, C. R., Shalizi, K. L., and Crutchfield, J. P., "An algorithm for pattern discovery in time series," Working Paper 02-10-060, Santa Fe Institute, 2003.

[294] Shenoy, R. P., "Artificial intelligence–the emerging technology," in *Defence Science Journal* [179], pp. 135–149.

[295] Silver, G. A., Lacy, L. W., and Miller, J. A., "Ontology based representations of simulation models following the process interaction world view," in *Proceedings of the 2006 Winter Simulation Conference* (Perrone, L. F., Wieland, F. P., J. Liu, B. G. L., Nicol, D. M., and Fujimoto, R. M., eds.), pp. 1168–1176, 2006.

[296] Simlote, A., "Joint warfare system (jwars)," in *Proceedings of the 2003 DoD User Group Conference*, pp. 325–329, June 2003.

[297] Simon, H. A., "How complex are complex systems?," in *Proceedings of the Biennial Meeting of the Philosophy of Science Association*, vol. 1976, pp. 507–522, 1976.

[298] Simpkin, R., *Deep Battle: The Brainchild of Marshal Tukhachevskii*. Potomac Books, 1987.

[299] Skorzeny, O., *My Commando Operations: The Memoirs of Hitler's Most Daring Commando*. Schiffer Publishing, 1995.

[300] Smith, E. A., *Effects-Based Operations: Applying Network Centric Warfare in Peace, Crisis, and War*. Command and Control Research Program Publications Series, 2003.

[301] Smith, E. A., *Complexity, Networking and Effects-based Approaches to Operations*. Command and Control Research Program Publications Series, 2006.

[302] Smith, G. L., "Sewanee and the environment: My vision," April 1995.

[303] Smith, R. D., "Essential techniques for military modeling & simulation," in *Proceedings of the 1998 Winter Simulation Conference* (Medeiros, D., Watson, E., Carson, J., and Manivannan, M., eds.), vol. 1, pp. 805–812, December 1998.

[304] Smuts, J. C., *Holism and Evolution*. New York: The MacMillan Company, 1926.

[305] Soban, D. S., *A Methodology for the Probabilistic Assessment of System Effectiveness as Applied to Aircraft Survivability and Susceptibility*. PhD thesis, Georgia Institute of Technology, December 2001.

[306] Sokolowski, J. A., "Enhanced military decision modeling using a multi-agent system approach," in *Proceedings of the Twelfth Conference on Behavior Representation in Modeling and Simulation*, pp. 179–186, May 2003.

[307] Solomonoff, R. J., "A formal theory of inductive inference," *Information and Control*, vol. 7, pp. 224–254, 1964.

[308] Spielman, D. A., "Spectral graph theory and its applications," in *48th Annual IEEE Symposium on Foundations of Computer Science*, (Providence, RI), pp. 29–38, 2007.

[309] Standish, R. K., "On complexity and emergence," *Complexity International*, vol. 9, pp. 1–6, 2001.

[310] Steels, L., "Towards a theory of emergent functionality," in *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, (Cambridge, MA, USA), pp. 451–461, MIT Press, 1990.

[311] Sterman, J. D., *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill/Irwin, 1st ed., February 2000.

[312] Stevens, W. K., "Use of modeling and simulation (m&s) in support of the quantitative assessment of forcenet systems and concepts," in *Proceedings of the 8th International Command and Control Research and Technology Symposium*, 2003.

[313] Stewart, I., "Networking opportunity," *Nature*, vol. 427, pp. 601–604, February 2004.

[314] Stone, G. F. and McIntyre, G. A., "The joint warfare system (jwars): A modeling and analysis tool for the defense department," in *Proceedings of the 33rd Winter Simulation Conference*, pp. 691–696, 2001.

[315] Suh, N. P., *Complexity: Theory And Applications*. Oxford University Press, 2005.

[316] Sulistio, A., Yeo, C. S., and Buyya, R., "A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools," *Software - Practice and Experience*, vol. 34, pp. 653–673, April 2004.

[317] Suppes, P., "Some remarks about complexity," in *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, vol. 2, pp. 543–547, 1976.

[318] Tarakanov, K. V., *Mathematics and Armed Combat*. Moscow: Military Publishing House, 1974.

[319] Ternion Corporation, "FLAMES Overview." Online at <http://www.ternion.com/>. Last accessed March 10, 2009.

[320] Thadakamalla, H. P., Raghavan, U. N., Kumara, S., and Albert, R., "Survivability of multiagent-based supply networks: A topological perspective," *IEEE Intelligent Systems*, vol. 19, no. 5, pp. 24–31, 2004.

[321] Tovar, E., "USMC Distributed Operations," in *Proceedings of DARPA Tech 2005*, vol. 22, pp. 22–24, DARPA, August 2005.

[322] Tsvetovat, M. and Latek, M., "Dynamics of agent organizations: Application to modeling irregular warfare," in *Proceedings of the 9th International Workshop on Multi-Agent-Based Simulation*, May 2008.

[323] Tzu, S., *The Art of War*. Forgotten Books, 1963. Translated from the Chinese by Lionel Giles, M.A. (1910).

[324] United States Air Force, "Modeling and simulation resource repository." Online at <https://afmsrr.afams.af.mil/>, February 2009.

[325] United States General Accounting Office, "Models, data, and war: A critique of the foundation for defense analyses," report to congress, United States General Accounting Office, Washington, DC, 1980.

[326] United States Government Office of Management and Budget of the Executive Office of the President of the United States of America, *The Budget for Fiscal Year 2008: Historical Tables*. U.S. Government Printing Office, 2007.

[327] Vacante, R. A., "Sense and Respond: An Emerging DoD Concept for National Defense," *Defense Acquisition Review Journal*, vol. 14, pp. 249–259, February 2007.

[328] Valverde, S. and Solé, R. V., "Internet's critical path horizon." 2004-06-010, June 2004.

[329] Varela, F. G., Maturana, H. R., and Uribe, R., "Autopoiesis: the organization of living systems, its characterization and a model," *Biosystems*, vol. 5, no. 4, pp. 187–196, 1974.

[330] Venkateswaran, J. and Son, Y.-J., "Distributed and hybrid simulations for manufacturing systems and integrated enterprise," in *Proceedings of the Annual Industrial Engineering Research Conference 2004*, (Houston, USA), May 15-19 2004.

[331] Volovoi, V., "Modeling of system reliability using petri nets with aging tokens," *Reliability Engineering and System Safety*, vol. 84, pp. 149–161, 2004.

[332] von Bertalanffy, L., "An outline of general system theory," *The British Journal for the Philosophy of Science*, vol. 1, pp. 134–165, August 1950.

[333] Wagenhals, L. W., Shin, I., Kim, D., and Levis, A. H., "C4ISR architectures: II. A structured analysis approach for architecture design," *Systems Engineering*, vol. 3, pp. 248–287, November 2000.

[334] Wang, W. and Brooks, R. J., "Improving the understanding of conceptual modelling," in *Proceedings of the 2006 OR Society Simulation Workshop*, 2006.

[335] Washburn, A., "Lanchester systems," April 2000.

[336] Watts, C. J., "Combining social network analysis with simulation modelling," in *Proceedings of the 2006 OR Society Simulation Workshop*, 2006.

[337] Watts, D. J. and Strogatz, S. H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, June 1998.

[338] Weaver, W., "Science and complexity," *American Scientist*, vol. 36, no. 536, p. 536, 1948.

[339] Weber, S. and Porto, M., "Generation of arbitrarily two-point-correlated random networks," *Physical Review E*, vol. 76, p. 046111, October 2007.

[340] Wilensky, U., "NetLogo," 1999. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL. Online at <http://ccl.northwestern.edu/netlogo>.

[341] Wilhelm, T., "An elementary dynamic model for non-binary food webs," *Ecological Modelling*, vol. 168, pp. 145–152, 2003.

[342] Wilson, B. and Fox, D. B., *Ground Combat in the JICM*. Rand Corporation, 1995.

[343] Wilson, E. O., *Consilience: The Unity of Knowledge*. New York: Alfred A. Knopf, Inc., 1998.

[344] Wiper, M., Pettit, L., and Young, K., "Bayesian Inference for a Lanchester Type Combat Model," *Naval Research Logistics*, vol. 47, no. 7, pp. 541–558, 2000.

[345] Wolfram, S., "Cellular automata as models of complexity," *Nature*, vol. 311, pp. 419 – 424, 4 October 1984.

[346] Wolfram, S., *Cellular Automata and Complexity: Collected Papers.* Addison-Wesley, 1994.

[347] Wolfram, S., *A New Kind of Science.* Wolfram Media, 2002.

[348] Womack, J. K., "Soviet correlation of forces and means: Quantifying modern operations," Master's thesis, Army Command and General Staff College, 1990.

[349] Wooldridge, M. and Jennings, N. R., "Intelligent agents: Theory and practice," *Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[350] Wuchty, S. and Stadler, P. F., "Centers of complex networks." 2002-09-052, September 2002.

[351] Wuensche, A., *Attractor Basins of Discrete Networks: Implications on Self-Organisation and Memory.* PhD thesis, The University of Sussex, 1996.

[352] Wuensche, A., "Discrete dynamical networks and their attractor basins," Working Paper 98-11-101, Santa Fe Institute, 1998.

[353] Yuhaski, S. J., "Multiechelon Network Model and Heuristic for the Combat Service Support Supply System (CS4)," Technical Report NATICK TR-94/030, United States Army Natick Research, Development and Engineering Center, August 1994.

[354] Zeigler, B. P. and Mittal, S., "Enhancing DoDAF with a DEVS-Based System Lifecycle Development Process," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3244–3251, October 10-12 2005.

[355] Zinn, A. W., "The Use of Integrated Architectures to support Agent Based Simulation: An Initial Investigation," Master's thesis, Air Force Institute of Technology, 2004.